



# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par **l'Institut Supérieur de l'Aéronautique et de l'Espace**  
Spécialité : Informatique

---

Présentée et soutenue par **Christophe HURTER**  
le 8 juillet 2010

**Caractérisation de visualisations et exploration interactive  
de grandes quantités de données multidimensionnelles**

---

### JURY

Mme Florence Sèdes, présidente  
M. Renaud Blanch  
M. Stéphane Conversy, directeur de thèse  
M. Jean-Daniel Fekete, rapporteur  
M. Philippe Palanque, directeur de thèse  
M. Guy Melançon

---

École doctorale : **Aéronautique - Astronautique**

Unité de recherche : **IRIT UMR 5505**

Directeur de thèse : **M. Philippe Palanque**  
Co-directeur de thèse : **M. Stéphane Conversy**

Autre rapporteur : **M. Irani Pourang**

## Table des matières

<b>Table des matières .....</b>	<b>2</b>
<b>Introduction .....</b>	<b>4</b>
<b>Partie 1 : Etat de l'art .....</b>	<b>9</b>
I. Le domaine de l'InfoVis .....	10
A. Les domaines de la visualisation .....	10
B. Exemples remarquables.....	11
II. La perception .....	14
A. La perception pré-attentive .....	14
B. Gestalt .....	15
C. La perception visuelle dans la sémiologie graphique de J. Bertin .....	16
D. Pouvoir perceptif .....	20
E. L'approche écologique de la perception visuelle .....	24
III. Analyse des visualisations .....	28
A. Le contenu.....	28
B. La mise en forme.....	34
IV. Evaluation des visualisations.....	47
A. Evaluation par les règles de design .....	47
B. Autres méthodes d'évaluation .....	50
V. L'exploration de données et le modèle de Data Flow .....	52
A. Le modèle de Data Flow pour l'exploration de données .....	52
B. La visualisation des données pour l'exploration.....	53
C. Interaction avec les données.....	56
D. Conclusion sur l'exploration de données.....	58
<b>Partie 2 : L'analyse de visualisations.....</b>	<b>59</b>
I. Caractérisation de visualisations dans le domaine du Contrôle Aérien.....	60
A. Caractérisation des visualisations du domaine de l'ATC.....	60
B. Image radar : ODS .....	62
C. Image radar verticale : ASTER .....	63
D. Séquenceur des avions à l'atterrissage : MAESTRO .....	65
E. L'agenda des conflits : ERATO .....	67
F. Discussion sur la caractérisation .....	67
G. La fonction de transformation généralisée.....	70
H. Le temps et la dynamique de l'image .....	71
I. Utilisation du modèle de la dynamique de l'image .....	73
J. Validation et utilisation du modèle de DataFlow .....	77
K. Conclusion .....	77
II. Comparaison des comètes radar.....	80
A. Les comètes radars dans le domaine de l'ATC.....	80
B. Description de quatre designs de comète.....	81
C. Application du modèle de C&M sur les comètes.....	83
D. Comparaison des comètes avec des critères qualitatifs .....	88
E. Conclusion sur la caractérisation des comètes radar .....	92
III. Design écologique et émergence .....	93
A. Définition de l'émergence .....	93

B.	Illustration du phénomène d'émergence .....	94
C.	Représentation utilisant le mouvement .....	99
D.	Discussion sur l'utilisation de la perception écologique.....	101
E.	L'émergence due à l'animation .....	102
F.	Exemple d'utilisation du design écologique.....	106
G.	Conclusion sur l'utilisation de l'écologie .....	110
<b>Partie 3 : La réalisation de visualisations .....</b>		<b>112</b>
I.	Réalisation de vues « métro » pour les routes aériennes .....	113
A.	L'analyse en type de donnée des vues métro et carte de routes aériennes .....	113
B.	Choix des couleurs et affectation sur les routes aériennes.....	119
C.	Visualisations réalisées .....	122
D.	Validation des visualisations .....	125
E.	Conclusion .....	127
II.	Interaction avec les données .....	128
A.	L'exploration de données dans le domaine de l'Aviation Civile.....	128
B.	Cahier des charges pour l'exploration de trajectoires.....	129
C.	Les données radar à explorer .....	129
D.	Description du système .....	130
E.	Premier bénéfice de l'utilisation de FromDaDy .....	134
F.	Scénario .....	137
G.	Utilisation étendue avec les cartes d'accumulation .....	141
H.	L'accumulation comme un outil d'exploration de données .....	147
I.	Considérations techniques .....	149
J.	Conclusion sur FromDaDy .....	151
<b>Conclusion.....</b>		<b>153</b>
<b>Références .....</b>		<b>157</b>
<b>Annexe.....</b>		<b>166</b>
I.	Validation avec le modèle de DataFlow.....	167
A.	Implémentation du modèle de DataFlow .....	167
B.	Exemple de visualisation : Minard .....	170
C.	Application du modèle de Data Flow aux visualisations de l'ATC.....	172
D.	Conception de nouveaux designs .....	180
E.	Conclusion .....	183



## Introduction

Un grand nombre d'activités utilise des outils informatisés s'appuyant sur des représentations graphiques interactives. Les représentations graphiques sont des vecteurs de transmission d'informations, visant à optimiser la bande passante entre le visuel et l'utilisateur. La représentation graphique d'informations doit assurer, selon Bertin [Bertin, 1983], trois fonctions principales :

- *enregistrer* de l'information (par exemple avec les plans de métro qui mémorisent toutes les connexions, les noms des stations...),
- *communiquer* de l'information (par exemple avec des outils comme PowerPoint ou les dessins à vocation pédagogique),
- *traiter* de l'information et résoudre des problèmes (par exemple avec une série temporelle des variations de la Bourse).

Aujourd'hui, pour la création de visualisations efficaces, les concepteurs utilisent plusieurs approches :

- **Historique** : le designer peut réutiliser des designs déjà existants en les adaptant dans un nouveau contexte. Cette démarche pose un problème, car le concepteur n'a pas les moyens lui permettant de connaître l'impact de ces modifications sur la perception des visualisations modifiées. Il ne peut donc pas garantir que le nouveau design conserve les propriétés du précédent.
- **Technologique** : le concepteur peut utiliser les capacités des nouvelles technologies et ainsi afficher de nouvelles informations par ce biais avec par exemple un écran 3D ou une résolution étendue de couleurs...
- **Empirique** : Cette méthode est la plus fréquemment utilisée. Le concepteur évalue son design avec des utilisateurs et le modifie en conséquence. Le processus de création est long et nécessite plusieurs itérations. De plus, l'évaluation des designs reste fortement liée à la qualité et au nombre d'utilisateurs.

Ces méthodes de production sont limitatives. Elles sont longues et ne reposent pas sur des fondements théoriques. Il n'existe pas de métrique commune pour l'évaluation des visualisations. Le concepteur perd beaucoup de temps à évaluer des designs avec les utilisateurs, qui ne sont pas toujours disponibles. Tous ces éléments altèrent le processus de design.

Le designer a besoin d'outils lui permettant d'évaluer la quantité et la qualité des informations transmises à l'utilisateur. Pour évaluer l'efficacité d'une visualisation, il faut au moins répondre aux questions suivantes : « Quelles sont les informations disponibles dans cette visualisation ? », « Quelles sont les informations perçues par l'utilisateur ? », « La visualisation instrumente-t-elle correctement l'activité ? ». Répondre à ces questions nécessite de décrire de façon systématique les visualisations.

Dans le domaine de la visualisation d'informations (InfoVis), il existe quelques outils qui se consacrent à l'étude des visualisations au travers de leur description. Tweedie [Tweedie, 1997] a démontré son utilité avec la réalisation d'une taxonomie des externalisations (visualisations). Ainsi, un nouveau domaine de l'espace de design a été

---

trouvé. Card et Mackinlay [Card & Mackinlay 1997] ont eux aussi porté leur intérêt sur la caractérisation, en proposant un outil permettant de décrire l'espace de design des visualisations du domaine de l'InfoVis.

### **La caractérisation**

La *caractérisation* de visualisations est une méthode qui permet la description complète ou partielle d'une visualisation. Dans sa définition la plus usuelle, le terme « caractériser » signifie : « définir précisément avec des caractères distinctifs »<sup>1</sup>. Afin de caractériser, il faut disposer d'un ensemble de concepts, d'éléments pour décrire. On parle alors de « nomenclature » : « une nomenclature désigne l'ensemble des termes propres à un domaine défini, par exemple une instance de classification (tableau, listes, règles d'attribution d'identité...) »<sup>1</sup>. L'exploitation directe d'une caractérisation permet la « classification » : « distribution systématique en diverses catégories, d'après des critères précis »<sup>1</sup>. L'ensemble des éléments ainsi classifiés forme une « taxonomie » (ou une taxinomie) : « Suite d'éléments formant des listes concernant un domaine »<sup>2</sup>. La taxonomie peut intégrer des relations entre ses éléments, on parle alors d'« ontologie » : « Une ontologie est un réseau sémantique qui regroupe un ensemble de concepts décrivant complètement un domaine. Ces concepts sont liés les uns aux autres par des relations taxonomiques (hiérarchisation des concepts) d'une part, et sémantiques d'autre part »<sup>1, 2</sup>. En résumé, la caractérisation de visualisations nécessite une description avec une nomenclature. L'ensemble des descriptions peut permettre de former une taxonomie, ou une ontologie si les descriptions ont des relations sémantiques entre elles.

Si l'on reprend la première définition, la caractérisation *de visualisations* consiste en la définition précise des visualisations, avec des caractères distinctifs. L'objectif d'une telle caractérisation est d'énumérer les propriétés des visualisations, afin de pouvoir comparer des visualisations similaires.

Nous pensons en effet que comprendre les ressemblances et les différences (y compris et surtout subtiles) entre les visualisations est une étape nécessaire à la formation d'un corpus de connaissances comme outil d'aide à la conception de visualisation. De tels outils de caractérisation permettront de synthétiser les designs sous une forme compacte, de comparer les descriptions entre designs, et d'instrumenter les dialogues entre concepteurs et utilisateurs avec un langage et une sémantique de description commune. Or une telle démarche de description et de comparaison est, à notre connaissance, embryonnaire.

La thèse que nous soutenons est que la caractérisation des visualisations est importante pour la conception de visualisation, et que son opérationnalisation par des méthodes et des outils permettent de concevoir de meilleurs systèmes interactifs.

---

<sup>1</sup> Le petit robert 2006

<sup>2</sup> [www.wikipedia.fr](http://www.wikipedia.fr)

---

## Méthodologie employée

Cette thèse utilise le trafic aérien (Air Traffic Management, ATM) comme domaine d'application. Nous avons rassemblé un ensemble de systèmes utilisant des représentations graphiques du domaine de l'ATM pour couvrir un large spectre de l'espace de design. L'image suivante montre un ensemble de visualisations du domaine ATC (Air Traffic Control) qui affichent toutes les mêmes informations : positions passées, courantes et futures des avions, routes aériennes et prévisions de conflits potentiels. Dans le domaine ATC, les visualisations peuvent être subtilement différentes ou très dissemblables. Elles évoluent régulièrement par ajout de nouvelles fonctionnalités : afin de vérifier qu'une modification n'altère pas les propriétés visuelles existantes, il est très important de pouvoir les décrire de façon correcte et complète.



En parallèle, nous avons réalisé un état de l'art des méthodes et outils disponibles dans des disciplines connexes pour caractériser les visualisations. Nous nous doutions en effet que les systèmes ATM pouvaient être considéré comme des visualisations. Nous avons ensuite appliqué ces outils aux visualisations de l'ATM, et notamment au contrôle aérien (ATC). Cette démarche nous a permis, entre autres, de valider que les outils conceptuels de domaines connexes sont applicables aux visualisations ATC, et qu'en ce sens le corpus de connaissances des disciplines connexes peut aider la conception des systèmes ATC.

Cependant, ces outils conceptuels ne sont pas suffisants pour caractériser la richesse des visualisations ATM. Ceci nous a conduit à modifier et étendre ces outils pour qu'ils intègrent des spécificités des visualisations riches et dynamiques, qu'ils tiennent compte de phénomènes perceptifs de haut-niveau comme l'émergence, ou le caractère écologique de leurs propriétés. Nous nous sommes appuyés en partie sur ces outils pour concevoir une nouvelle visualisation pour le contrôle aérien, basé sur une vue de type Metro.

---

Afin d'évaluer la correction de notre analyse des visualisations ATC, nous avons opérationnalisé le modèle de caractérisation dans un logiciel, notamment la caractérisation par flot de données, et la caractérisation de l'accumulation. Le logiciel recrée les entités visuelles principales qui composent les visualisations étudiées, et il nous a permis de vérifier que la visualisation produite est conforme à l'originale.

Le logiciel réalisé nous a permis d'une part de comprendre le processus de transformation de données vers une visualisation, et d'autre part d'interagir avec les données affichées et ainsi procéder à leur exploration. En effet, ce logiciel est basé sur une visualisation interactive de données avec des choix de représentations configurables pour amplifier la cognition. Notre démarche de validation a donc donné lieu à la réalisation d'un logiciel capable de faire de l'exploration interactive de données. Un utilisateur de ce logiciel peut d'une part modifier les choix de design pour optimiser la perception d'information et, d'autre part, interagir avec les données en procédant à des filtrages successifs à l'aide d'un étalement progressif des données dans des vues juxtaposées. C'est cette combinaison entre un système de visualisation et des paradigmes d'interaction qui permet de réaliser des requêtes complexes sur les données.

### **Organisation du mémoire**

**Partie 1** – « L'état de l'art » contient les bases théoriques sur lesquelles nous nous appuyons tout au long de notre recherche. De plus, cette partie contient notre analyse des travaux fondateurs avec, d'après notre expérience, les manques potentiels.

**Partie 2** – « L'analyse de visualisations » est composée de l'analyse des visualisations du domaine de l'ATC et la réalisation de visualisations plus efficaces.

**Partie 3** – « La réalisation de visualisations » est axée sur la conception de visualisations efficaces : des vues de type « métro » pour les routes aériennes, et un logiciel pour explorer de grandes quantités de données.

Par ailleurs, l'annexe comporte les détails de notre implémentation du modèle de DataFlow pour la validation du processus de caractérisation et pour l'exploration de données. Cette partie ne contient pas de contribution mais permet de valider des concepts importants à mettre en œuvre pour, par exemple, réaliser la vue radar des Contrôleurs Aériens.

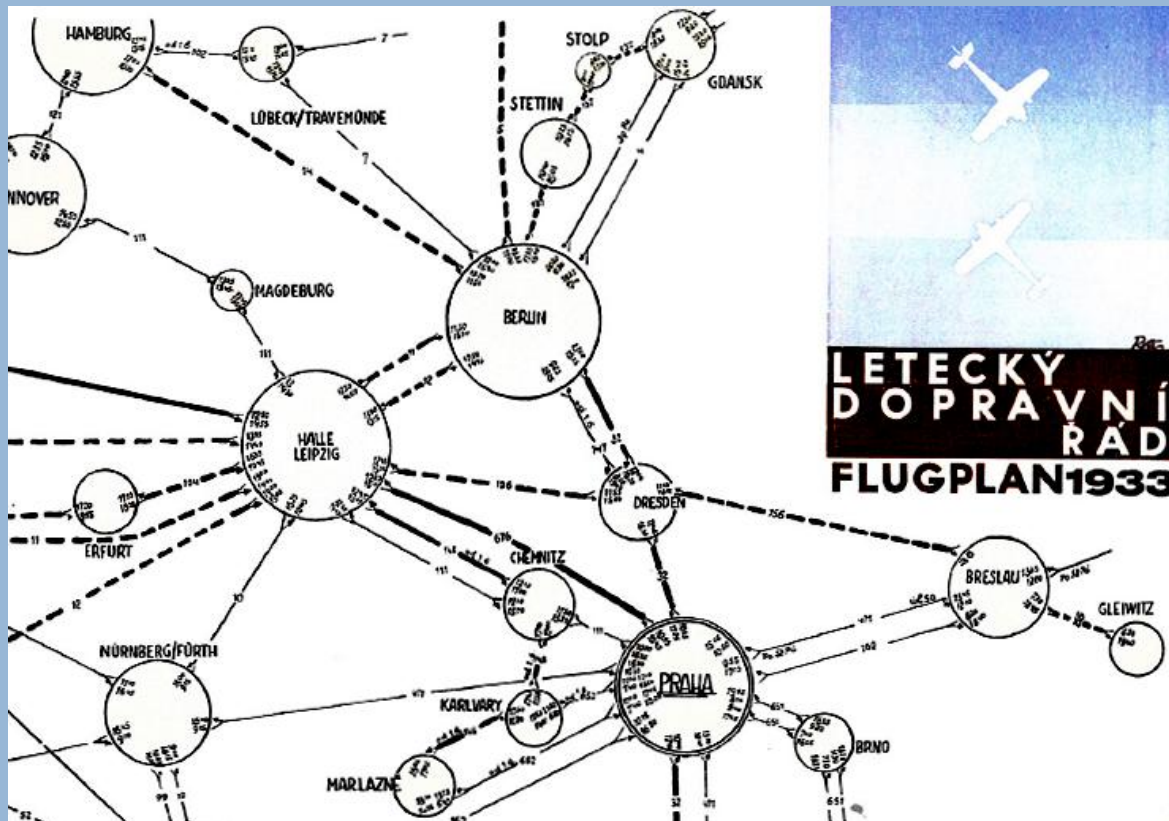
---

### **A l'attention du lecteur**

Comme l'ont fait Bertin ou Ware [Ware, 2008], les légendes des images ont volontairement été omises. Les contraintes de mise en page sont à l'origine du besoin de référencement des images, or dans cette thèse, le texte précède toujours l'image qu'il décrit. De plus, elles sont souvent auto-suffisantes à leur compréhension.



## Partie 1 : Etat de l'art



Visualisation des horaires des compagnies aériennes, Tchécoslovaquie 1933  
[Tufté, 1990]

Cette partie, structurée en cinq sections, décrit les travaux antérieurs en relation avec nos travaux de recherche. Après une introduction sur l'InfoVis pour situer nos travaux, nous décrivons les modèles de perception qui expliquent les mécanismes de haut niveau mis en œuvre lors de la transmission d'information d'un visuel vers un utilisateur. Ces modèles constituent la base des outils de caractérisation. Nous continuons avec les travaux relatifs à la caractérisation des visualisations, à la mise en forme des informations issues de cette caractérisation et à l'évaluation des visualisations. Enfin cette partie se termine avec, d'une part, l'étude du modèle de flot de donnée, qui est l'élément principal pour la caractérisation et la réalisation de visualisations, et d'autre part, l'étude des outils pour l'exploration des données. Cette dernière partie est liée aux travaux que nous avons réalisés sur l'opérationnalisation de la caractérisation.

## I. Le domaine de l'InfoVis

---

Le domaine de la visualisation d'informations (InfoVis) est né avec l'évolution technologique des supports interactifs pour transmettre de l'information en utilisant des représentations abstraites. Parmi les travaux fondateurs du domaine on retrouve ceux de Tufte [Tufte, 1983] [Tufte, 1990] [Tufte, 1997] qui a regroupé un ensemble de visualisations affichant des informations scientifiques mais aussi abstraites. Bertin [Bertin, 1983] fait aussi partie des fondateurs avec *la sémiologie graphique*, ainsi que Tukey [Tukey, 1977] avec ses travaux sur la réalisation de visualisations de données statistiques, Cleveland et McGill [Cleveland, 1985] [Cleveland & McGill, 1988] [Cleveland, 1995] avec de nouvelles visualisations abstraites.

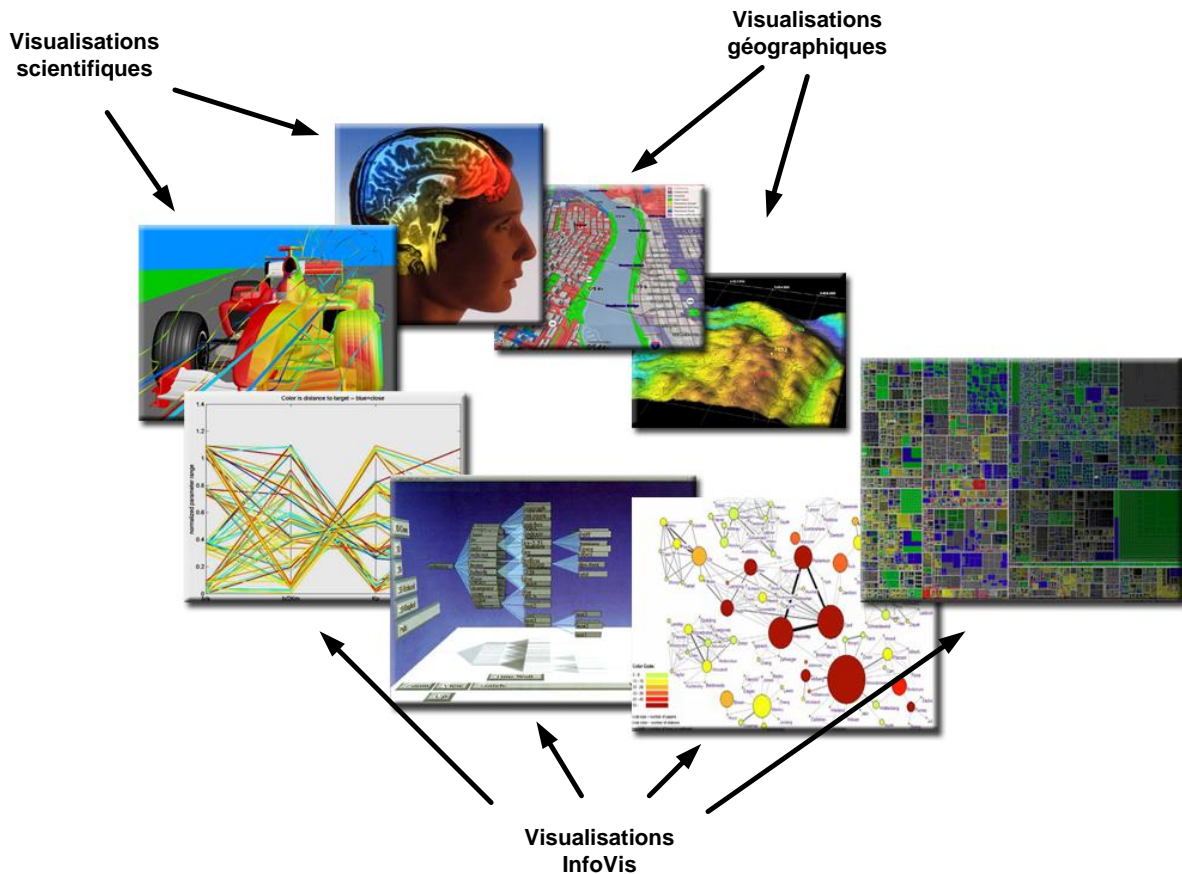
### A. Les domaines de la visualisation

---

On peut concevoir l'InfoVis comme une spécification et une évolution des domaines de la visualisation de données géographiques et scientifiques :

- La cartographie est un domaine vieux d'au moins 2000 ans. Les cartes permettent de mémoriser des informations géographiques. Ainsi, elles aident à se diriger mais aussi à trouver et à analyser de l'information ; il est ainsi possible de découvrir des zones non encore cartographiées, des zones de prospection... Aujourd'hui ce domaine est celui des Système d'Informations Géographiques (SIG).
- Les visualisations scientifiques (SciVis) ont une vingtaine d'années d'existence. Le domaine des visualisations scientifiques traite principalement des représentations « canoniques » de phénomènes physiques. Elles font toujours référence à des objets connus mais en ajoutant de l'information perceptible par notre système cognitif. Par exemple, l'image d'une voiture de course avec une représentation de l'écoulement de l'air sous forme de filets colorés est une visualisation du domaine scientifique.
- Le domaine de l'InfoVis a une dizaine d'années d'existence. Il affiche des données de façon abstraite sans référentiel dans notre environnement naturel mais en utilisant notre système perceptif comme vecteur d'amplification de la cognition.
- Le domaine de l'InfoVis continue d'évoluer avec l'émergence du nouveau domaine de l'analyse visuelle [Thomas & Cook, 2006]. L'analyse visuelle (Visual Analytics) est la science du raisonnement assisté par des interfaces visuelles interactives.

L'image suivante montre des exemples de visualisations du domaine de la cartographie avec des cartes et des reliefs de terrain, de la visualisation scientifique avec l'image de zones d'activation cérébrale ou encore une voiture de course avec l'affichage des filets d'air, de l'InfoVis avec Conetree [Roberstion et al., 1991], Parallel Coordinates [Inselberg & Dimsdale, 1990], Tree-maps [Johnson & Shneiderman, 1991].



La séparation entre les quatre domaines cartographique, scientifique (SciVis), InfoVis et Visual Analytics n'est pas toujours très marquée. Elle a été discutée dans un workshop [Rhyne et al., 2003] ce qui a permis de mieux définir la position de chercheurs influents du domaine. Malgré l'existence d'un fort recouvrement entre ces domaines, il est néanmoins possible de proposer une définition pour chacun d'entre eux :

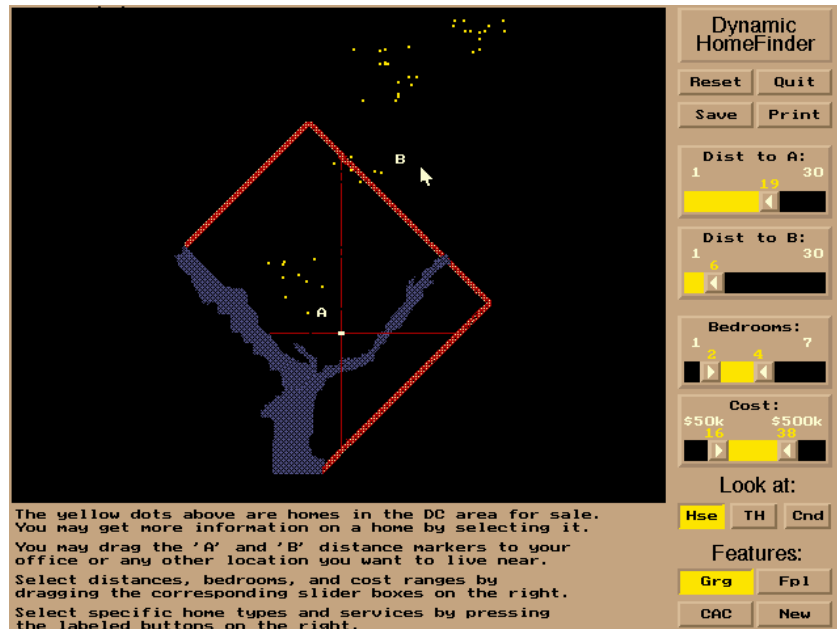
Cartographie	Représentation de données géographiques.
SciVis	Représentation et interaction avec des données scientifiques principalement basées sur des objets physiques qui permettent d'amplifier la cognition.
InfoVis	Utilisation interactive de représentations visuelles abstraites qui permettent d'amplifier la cognition.
Visual Analytics	Science du raisonnement assisté par des interfaces visuelles interactives et l'Intelligence Artificielle.

## B. Exemples remarquables

L'exemple suivant est une image statique de la marche de Napoléon lors de sa campagne de Russie en 1812. Cette image a été réalisée par Minard en 1869 et fait référence dans le domaine de l'InfoVis en affichant une représentation abstraite de données très simplement compréhensibles. Un trait qui se sépare en plusieurs branches représente la



Le dernier exemple utilise la technique d'interaction des requêtes dynamiques qui permet de filtrer dynamiquement les données pour en extraire rapidement de l'information. HomeFinder permet de trouver une propriété à acheter suivant des critères manipulés directement par l'utilisateur. Pour ce faire, l'utilisateur utilise des range sliders et des cases à cocher pour définir et affiner ses critères de sélection.



L'utilisation simple et rapide d'outils de sélection permet alors d'étendre les possibilités d'exploration. L'utilisateur peut, en outre, trouver un logement mais aussi par extension, comprendre la topologie immobilière de la région de Washington en mettant en avant les zones fortement urbanisées, les zones impopulaires ou encore les quartiers résidentiels cossus. La requête (filtrage) suivie de sa visualisation immédiate permet à l'utilisateur de mémoriser et d'identifier des formes caractéristiques dans la visualisation des données et ainsi de les explorer.

## II. La perception

---

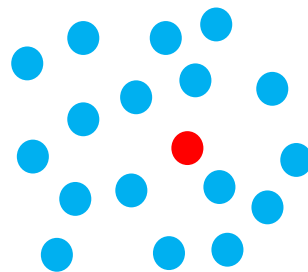
Cette partie résume quelques unes des théories de la perception visuelle et présente leur utilisation dans le domaine de la visualisation d'information. En revanche, cette thèse ne s'inscrit pas dans le domaine de l'étude des processus physiologiques mis en œuvre pour la perception d'informations. Nous nous intéressons aux théories et modèles de la perception de « haut niveau » pour pouvoir expliquer les mécanismes de transmission d'information entre une visualisation et un utilisateur.

Nous commençons par l'étude de la perception pré-attentive qui permet d'identifier rapidement certaines propriétés visuelles simples. Nous présentons ensuite la théorie de la Gestalt pour laquelle la perception des objets dans leur ensemble est supérieure à la perception individuelle. Ensuite nous étudions la sémiologie graphique de Bertin qui offre un socle formel pour l'analyse des visualisations. Puis, nous nous intéressons à l'approche du design écologique qui décrit la perception comme un système fortement lié à l'environnement naturel de l'Homme. Finalement, nous étudions le pouvoir perceptif des choix de représentation.

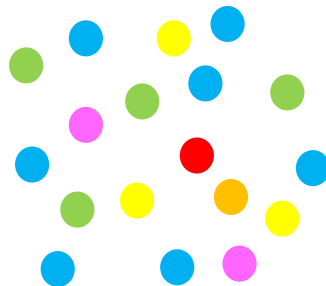
### A. La perception pré-attentive

---

La théorie de la perception pré-attentive [Treisman, 1988] explique comment certaines propriétés visuelles peuvent être perçues très rapidement (< 250 ms) et sans effort. Par exemple, lorsque l'on regarde l'image suivante, composée de cercles bleus, la perception du cercle rouge est immédiate et ne requiert aucune charge cognitive : ce processus de perception est automatique.



Cette perception est hautement parallèle mais limitée : la perception automatique du cercle rouge ne fonctionne plus avec un ensemble plus important de couleurs. L'utilisateur est alors obligé de scanner chaque élément de façon séquentielle.



La couleur, l'orientation, la longueur, l'épaisseur, la courbure, et d'autres encore peuvent participer à la perception pré-attentive [Healey, 1992]. A l'opposée de cette perception automatique, la perception contrôlée concerne, par exemple, la lecture d'un texte. Cette

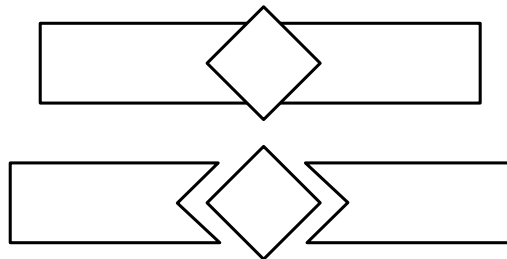


activité mobilise entièrement le système cognitif, et ne laisse pas de place à d'autres activités perceptives. Les informations transmises par la lecture d'un texte sont précises, mais leur temps de traitement est très lent.

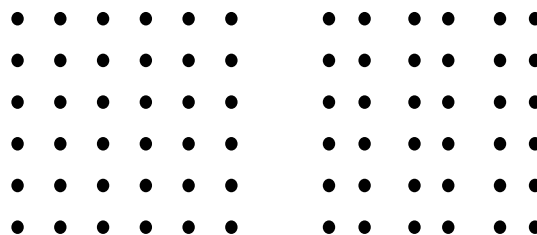
## B. Gestalt

La théorie de la Gestalt [Kofka, 1935] est née en Allemagne et en Autriche au 19<sup>ième</sup> siècle. Le mot allemand « Gestalt » signifie « forme ». Cette théorie part du fait que la perception des objets est réalisée par la synthèse mentale des formes en une « surforme », c'est un processus de haut niveau. On constate ainsi que le tout est supérieur à la somme de ses parties, c'est un des principes fondamentaux de la théorie de la Gestalt. La Gestalt regroupe plusieurs principes de la perception : la loi de la bonne forme, la loi de continuité, la loi de proximité, la loi de similitude, la loi du destin commun...

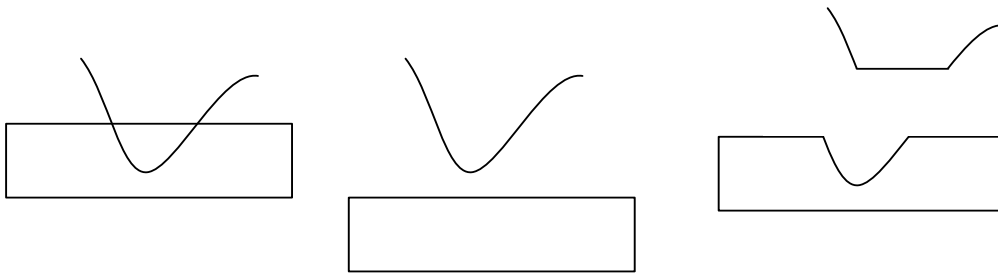
La loi de la *bonne forme* (Prägnaz) est la loi principale dont les autres découlent : un ensemble de parties (comme des groupements aléatoires de points) tend à être perçu d'abord (automatiquement) comme une forme, cette forme se voulant « simple », « symétrique », « stable », en somme une « bonne » forme. L'exemple suivant montre un rectangle surmonté d'un carré : le système perceptif préfère cette composition de l'espace qui est plus simple que, par exemple, trois objets distincts juxtaposés.



La loi de la *proximité* groupe naturellement les objets les plus proches les uns des autres. Le carré de gauche est composé de points uniformément distants, alors que dans carré de droite, les points se regroupent en colonne par effet de proximité.



Le principe de la loi de *continuité* statue qu'il est plus facile de construire des entités visuelles à partir d'éléments qui sont lissés et continus plutôt qu'avec des éléments qui comportent des angles, ou des changements de direction abrupts. L'image suivante montre que nous percevons une courbe et un rectangle superposés, et non deux éléments juxtaposés.

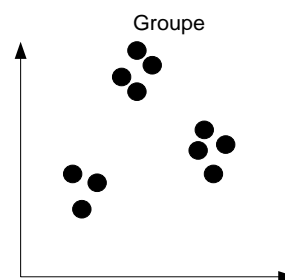
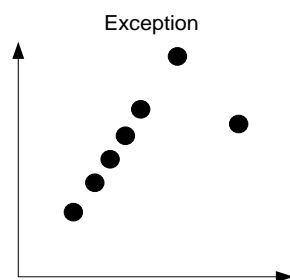
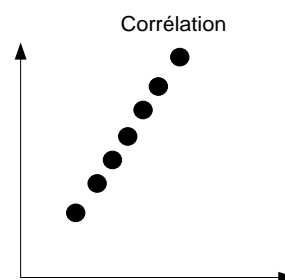
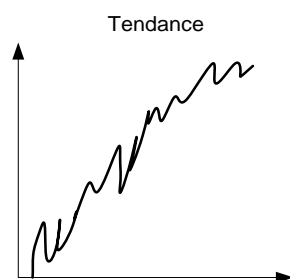


La loi de *similitude* indique que les éléments visuels qui partagent les mêmes propriétés se groupent. Par exemple des points bleus se groupent au milieu de point gris.

La loi du *destin commun* : des parties en mouvement ayant la même trajectoire sont perçues comme faisant partie de la même forme.

La loi de *clôture* : une forme fermée est plus facilement identifiée comme une figure (ou comme une forme) qu'une forme ouverte.

A titre d'exemple, la Gestalt permet d'expliquer les processus perceptifs mis en œuvre dans la perception des informations contenues dans un scatterplot. Le traitement hautement parallèle du système visuel humain permet de percevoir sans surcharge cognitive des tendances (loi de la continuité), des patterns (loi de la similitude), des exceptions, des groupes (loi de la proximité).



## C. La perception visuelle dans la sémiologie graphique de J. Bertin

---

La sémiologie graphique de Bertin [Bertin, 1983] apporte une réponse objective aux questions suivantes : « dans quel cas faut-il faire un dessin ? », « Quel dessin faut-il faire ? », « Que contient le dessin ? ». Pour répondre à ces questions, Bertin a défini un



formalisme rationnel qui permet d'une part de décrire avec précision les visualisations, et d'autre part de comprendre pourquoi l'utilisateur perçoit les informations qu'elles contiennent.

Bertin a introduit la *sémiologie graphique* qui est un système monosémique de représentation graphique pour transmettre de l'information sans ambiguïté. Il a défini les sept variables visuelles qui forment les éléments élémentaires pour la composition des images. Et enfin, il a défini les propriétés perceptives de chacune des variables visuelles.

## 1. Les variables visuelles de Bertin

Bertin est le premier à avoir proposé, de façon formelle, un ensemble fondamental de variables visuelles qui sert à la construction élémentaire de toute représentation graphique. La sémiologie graphique décrit sept variables visuelles qui forment les éléments de plus bas niveau pour la composition graphique [Bertin 1983]. Il y a la position (la variable visuelle spatiale), la taille, la valeur (proche de la luminosité), le grain, la couleur (mélange de teinte et de saturation), l'orientation, la forme. Ces variables visuelles sont appliquées à trois types d'implantation dans le plan : le point, la ligne, la zone.

Variables Visuelles (Vv)	
Position X	
Position Y	
Taille	
Valeur	
Grain	
Couleur	
Orientation	
Forme	

Bertin définit la variable visuelle *grain* comme la notion utilisée par les photographes. Dans ce cas, la granularité peut être associée à une fréquence spatiale dans une texture ; c'est pourquoi la variable visuelle *grain* est souvent appelée texture dans la littérature.

Les variables visuelles ont des échelles de perception différentes appelée *longueur*. La *longueur* d'une variable visuelle est le nombre d'éléments différenciables avec cette variable visuelle. Par exemple, la Variable Visuelle de position possède l'échelle de perception la plus longue et offre même une infinie possibilité de variation. Cette variation infinie est théorique puisque limitée par la taille du support (écran, feuille de papier...).

## 2. La perception d'information en fonction du type des variables visuelles de Bertin

Les Variables Visuelles ont chacune leur propre niveau d'organisation. Ainsi on ne peut pas percevoir un ordre si la variable visuelle n'est pas « ordonnée », et on ne peut percevoir un rapport si la variable n'est pas « quantitative ».

On associe aux variables visuelles quatre types de perception que Bertin appelle les « niveaux » des variables visuelles :

- une variable visuelle est *associative* si elle permet à une catégorie nominale d'être perçue malgré l'influence des autres variables dans une même image. Par exemple, être capable de trouver des formes triangulaires dans une image quelque soit leur couleur ou leur position,
- une variable visuelle est *sélective* lorsqu'elle permet à une catégorie nominale d'être perçue instantanément « en avant » dans l'image. Par exemple, l'ensemble créé à partir de tous les points bleus dans un nuage de points multicolores,
- une variable visuelle est *ordinale* lorsque le classement visuel de ses catégories est spontané et universel. Chacune permet à l'utilisateur de percevoir une gradation dans la valeur affichée (« ceci est plus grand ou plus petit que cela »). On perçoit un gris comme l'intermédiaire entre un blanc et un noir, une taille moyenne comme l'intermédiaire entre une petite et une grande taille. En revanche, les couleurs (par exemple bleu, rouge ou vert) n'offrent pas spontanément un ordonnancement,
- une variable visuelle est *quantitative* lorsque la distance visuelle entre les catégories d'une composante ordonnée peut s'exprimer spontanément par un rapport numérique. On perçoit qu'une longueur est égale à trois fois une autre longueur, qu'une surface est le quart d'une autre surface. L'échelle *quantitative* permet à l'utilisateur de juger la quantité qui différencie deux éléments.

## 3. Propriétés perceptives des variables visuelles

Bertin utilise une symbolique pour qualifier les niveaux des variables visuelles :

<b>Q</b>	Quantitative (proportionnelle)
<b>O</b>	Ordonnée
<b>≠</b>	Sélective (différentielle et instantanée)
<b>≡</b>	Associative (visibilité inchangée des autres Variables Visuelles)
<b>≠</b>	Dissociative (visibilité variable des autres Variables Visuelles)

Le tableau des correspondances entre les variables visuelles et leurs qualités perceptives permet de résumer le pouvoir de chaque variable visuelle avec une gradation sur leur efficacité (graisse des sigles, avec les éléments en gras les plus efficaces) :

	POINTS	LIGNES	ZONES				
XY 2 DIMENSIONS DU PLAN							
Z TAILLE							
VALEUR							
<b>LES VARIABLES DE SÉPARATION DES IMAGES</b>							
GRAIN							
COULEUR							
ORIENTATION							
FORME							

Voici quelques remarques suite à notre lecture de ce tableau :

Selon Bertin, une image qui utilise une correspondance Variable Visuelle / Niveau ( $Q, O, \neq, \neq$ ) s'inscrivant dans les cases vides du tableau est une « convention » ; cette convention détruit la signification des données car le choix du design de l'image n'est pas adapté à la transmission correcte d'information. Par exemple, la couleur ne peut pas être utilisée pour représenter des valeurs de distance entre avions (données de type quantitatives).

Bertin décrit le *plan* (i.e. X et Y) comme la plus riche des variables visuelles. Le plan est sélectif, associatif, ordonné et quantitatif. Bertin catégorise ces implantations avec le point, la ligne ou la surface. Le type d'implantation affecte la longueur des variables visuelles. Ainsi, l'implantation de zone fait émerger deux problèmes : l'orientation ne peut plus être perçue comme sélective, et la perception des autres variables visuelles est la même sur toute la zone (les quantités doivent être normalisées par unité de surface, sinon elles ne pourront pas être interprétées correctement).

La *taille* est la seule variable visuelle quantitative (elle permet de faire des rapports entre les quantités perçues). Elle est sélective et ordonnée mais non associative. La taille et la valeur (luminosité) sont dissociatives car elles ont une visibilité variable. C'est la contrepartie de leur fonction essentielle : créer l'image. Il n'est pas possible d'exclure cette variation de visibilité : les autres variables visuelles ont une visibilité constante et sont dites associatives.

La *valeur* est le ratio entre la perception du blanc et du noir sur une surface. Elle est ordonnée. L'utilisation de la valeur de Bertin est à rapprocher avec la luminosité du modèle des couleurs LCH (Teinte, Saturation, Luminosité). La taille et la valeur sont ordonnées. Mais comme le blanc ne peut pas servir d'unité pour mesurer le noir, la valeur n'est pas quantitative.

Bertin n'a pas créé de séparation entre la teinte et la saturation, mais la variable visuelle *couleur* est plus proche de la teinte que de la saturation. La couleur n'est pas implicitement ordonnée, mais elle est sélective et associative.

Le *grain* est la variation d'échelle des composantes constituant une texture. Le grain peut être considéré comme une composition d'éléments de même taille, il est alors ordonné. Il est à la fois sélectif et associatif. La longueur de cette variable est basée sur la taille de son implantation. Ainsi, l'utilisation de marques plus grosses permet une longueur plus importante.

L'*orientation* est l'angle relatif entre deux marques. L'orientation est associative mais n'est sélective que pour les points et les lignes. Elle n'est pas ordonnée. Il existe de multiples façons de diviser 360 degrés en étape. Néanmoins, Bertin définit seulement quatre étapes pour maximiser la sélectivité de l'orientation ; la longueur de la variable visuelle *orientation* est ainsi réduite.

La *forme* est la variable visuelle la plus ambiguë car elle mélange des aspects de l'*orientation* et de la *taille*. Elle n'est pas ordonnée. Elle a une infinité de longueur et est associative et très faiblement sélective.

L'utilisation des variables visuelles de Bertin est une façon de décrire les images. Il est possible de les appliquer au-delà des limites d'une feuille (avec ses deux dimensions) et ainsi d'ajouter d'autres variables visuelles comme la profondeur ou le temps pour l'animation. Les images en trois dimensions peuvent aussi être étudiées sous la forme d'une projection en 2 dimensions et ainsi rester dans le cadre d'une image dans un plan.

## **D. Pouvoir perceptif**

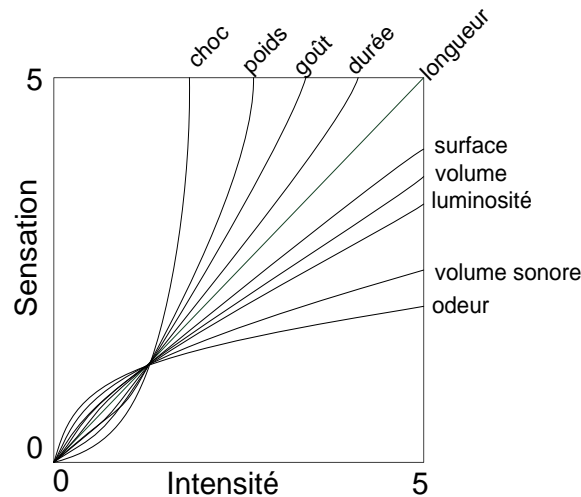
---

Dans cette partie, nous détaillons les travaux remarquables qui permettent de classer des propriétés perceptives en fonction de leur pouvoir perceptif.

### **1. Loi de Weber**

La loi de Weber permet d'évaluer la perception d'une stimulation en fonction de son intensité. Par exemple, la perception de l'intensité lumineuse varie en fonction du carré de la distance à la source. La perception des stimuli n'est pas une fonction linéaire entre l'intensité du stimulus et la sensation perçue. Weber puis Fechner ont étudié cette relation et en ont déduit une loi sous la forme  $S = k \log(I) + c$ , avec  $k$  et  $c$  deux constantes fonction du type de stimuli.

La figure suivante donne les courbes de correspondance entre l'intensité des stimuli et l'intensité de la perception en fonction de différents stimuli. Ces données sont principalement issues des expérimentations de Shiffman [Shiffman, 1990] et Stevens [Stevens, 1961].

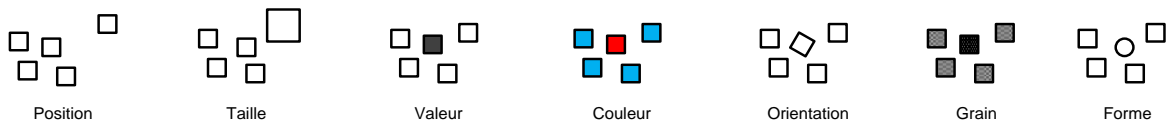


## 2. Le pouvoir sélectif des Variables Visuelles

Une variable est dite *sélective* si elle partitionne l'espace visuel. La perception sélective a lieu quand une variable visuelle fait ressortir une catégorie donnée de la visualisation. Elle permet d'isoler spontanément toutes les correspondances appartenant à une même catégorie (de cette variable) : « ceci est différent de cela ». Ces correspondances forment entre elles une « famille » : la famille des signes rouges, des signes verts... On peut alors parler de la sélection pré-attentive [Treisman, 1985] : capacité de distinguer un objet parmi d'autres sans tester chaque objet. Toutes les variables visuelles sont sélectives mais à différents degrés. La liste suivante donne, par ordre décroissant, le niveau de sélectivité des variables visuelles défini par Bertin [Bertin, 1983].

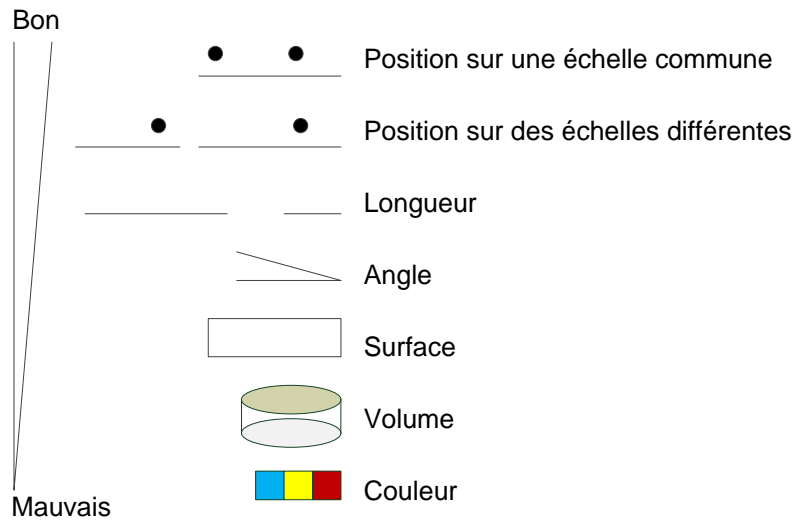


Pouvoir sélectif des Variables Visuelles



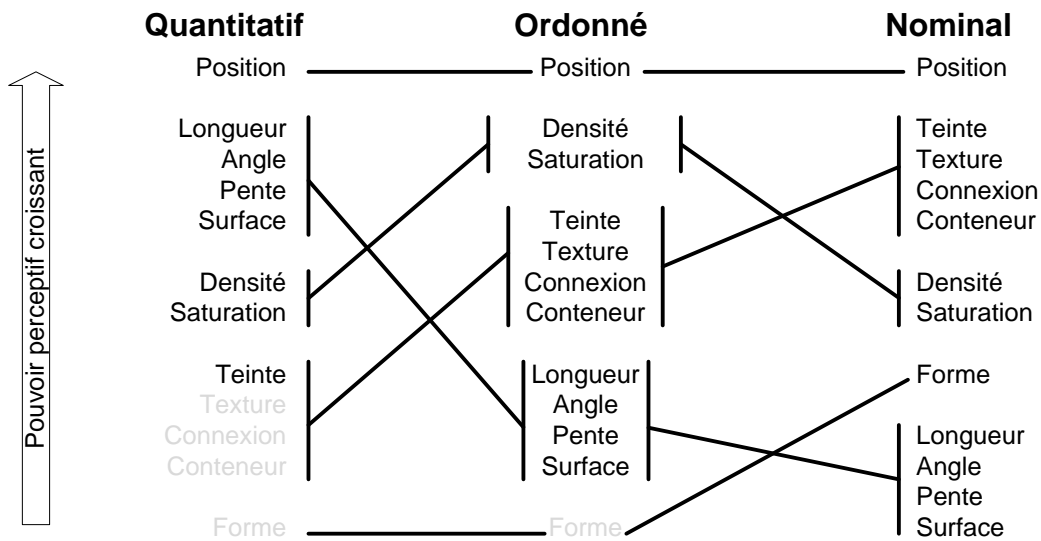
## 3. La classification de Cleveland & McGill

La classification de Cleveland [Cleveland, 1985] [Cleveland & McGill, 1988] a été réalisée pour évaluer la qualité des visualisations de données statistiques. Cleveland a évalué l'efficacité des variables visuelles pour représenter des informations de type intervalle (I) et ratio (Q). Cette évaluation a été réalisée avec des éléments visuels isolés de tout contexte. L'image suivante montre le classement qu'ils ont réalisé Cleveland.



#### 4. La classification de J. Mackinlay

Mackinlay [Mackinlay, 1986] a évalué par des expérimentations le pouvoir perceptif des variables visuelles en fonctions du type de données (Quantitatif, Ordonné, Nominal) à afficher. Ces évaluations ont été réalisées avec les données quantitatives. Les classements des données O et N sont réalisés par déduction et non par expérimentation. Ce diagramme permet de faire des choix de design avec le type de données à afficher.



Il est intéressant de noter que ce diagramme permet de résumer l'utilisation des variables visuelles en fonction du type de données à afficher. Il indique en plus l'évolution de l'efficacité des variables visuelles en fonction du type de données (Q, O, N). Les croisements des lignes lient les variables visuelles indiquent que les propriétés graphiques des variables visuelles sont très différentes en fonction des types de perception. A titre d'exemple, la taille est très efficace pour coder visuellement des données quantitatives et son efficacité décroît avec les types O et N. Seule la position est la variable visuelle la plus efficace quel que soit le type de données. On retrouve la position de Bertin sur ce sujet.

## 5. La classification de Wilkinson

Wilkinson a, comme Mackinlay, utilisé les variables visuelles de Bertin et les a classifiées en fonction de leur efficacité pour coder visuellement des données quantitatives [Wilkinson, 1999]. Il a étendu l'ensemble des variables visuelles en ajoutant des attributs visuels supplémentaires comme la saturation, le flou. Le tableau suivant présente les choix de design, ordonnés du plus efficace au moins efficace en fonction du vecteur de transmission d'information (la forme, la surface, l'animation, le son et le texte).

Forme	Surface	Animation	Son	Texte
Position	Couleur	Direction	Hauteur	Texte
Taille	Saturation	Vitesse	Volume	
Formes	Luminosité	Accélération	Rythme	
Polygone	Teinte		Voix	
Glyphes				
Image	Texture			
	Motif			
Rotation	Granularité			
Résolution	Orientation			
	Flou			
	Transparence			

De plus, Wilkinson a étudié l'animation, le son ou encore le texte comme vecteur d'informations. Il existe encore d'autres attributs « esthétiques » mais, selon Wilkinson, ils sont difficilement exploitables : par exemple l'odorat, ou le touché. Ceci s'explique par les résultats du domaine de la psychophysique et l'étude des relations entre l'amplitude d'un stimulus et sa perception [Stevens, 1985] [Falmagne, 1985]. La loi de Fechner/Weber permet de quantifier cette relation et ainsi de mettre en avant la notion de « Just Noticeable Difference » (JND) qui représente la différence minimale de stimulation pour la perception (on peut la rapprocher du terme de longueur des variables visuelles de Bertin mais étendu à tous les sens de la perception).

## 6. Extension des variables visuelles de Bertin

Bertin a été le premier à proposer des choix de design en fonction du type de données à afficher. Le tableau suivant est extrait de l'ouvrage « How maps works » de MacEachren [MacEachren, 1995] qui synthétise les travaux de Bertin dans ce domaine.

	Quantitatif	Ordinal	Nominal
Position			
Taille			
Valeur			
Texture			
Couleur			
Orientation			
Forme			

Utilisation acceptable	
Utilisation impossible	

MacEachren a étendu les variables visuelles et leur utilisation dans les cartes avec par exemple l'utilisation de la transparence, la résolution, la décomposition des couleurs en luminosité, teinte et saturation. Aujourd'hui, les designers utilisent couramment l'espace de couleur LCH. Morison [Morrison, 1974] pense que la Variable Visuelle couleur de Bertin peut être scindée en deux attributs : couleur et saturation. Bertin parle aussi de saturation des couleurs mais il a choisi de grouper cet attribut avec la teinte sous le terme de couleur.

## **E. L'approche écologique de la perception visuelle**

---

L'approche écologique de la perception est une explication du processus de capture d'informations de l'environnement que l'on doit à James Jerome Gibson, psychologue du XXème siècle [Gibson, 1977]. Elle remet en cause à la fois la notion d'informations à percevoir et la définition des systèmes perceptifs humains. Nous pensons que les concepts d'approche écologique de la perception, notamment ceux d'invariant et d'affordance, peuvent permettre d'expliquer en quoi une visualisation est meilleure qu'une autre. La suite de cette section présente de façon succincte cette approche, afin d'expliquer les concepts d'invariant et d'affordance que nous utilisons dans la partie 2.

### **1. Le système perceptuel visuel humain**

Une des notions clé de l'approche écologique de la perception consiste à redéfinir ce qu'il y a à percevoir, et les moyens humains pour percevoir. Selon les approches traditionnelles de la perception, l'acte de perception ne commence qu'à partir de l'image formée au fond de la rétine, c'est-à-dire d'une surface formée de points de couleurs et d'intensités différentes. Pour Gibson, l'œil n'est qu'une partie d'un système perceptuel, comprenant d'autres parties du corps humain, et participant au processus de vision. Notre système perceptuel visuel est constitué de deux yeux mobiles fichés dans une tête, la tête est elle-même mobile et peut tourner selon trois axes au niveau du cou, et un être humain peut se déplacer grâce à ses jambes.

En considérant un tel système, les informations à percevoir ne se résument plus à une image rétinienne. Grâce aux possibilités de mouvement des organes, l'être humain peut agir sur la perception : il peut bouger ses yeux, pencher la tête, avancer. Les informations visuelles deviennent dynamiques, l'image se transforme en flot continu d'images. De plus, une personne peut revenir à une position initiale, fixer un point ou revenir à un point précédent, avancer et reculer. Ces mouvements peuvent être commandés par une volonté de percevoir "mieux", c'est-à-dire que le processus perceptif n'est pas seulement un processus passif mais une boucle perception-action, où l'action génère la perception et la perception commande l'action.

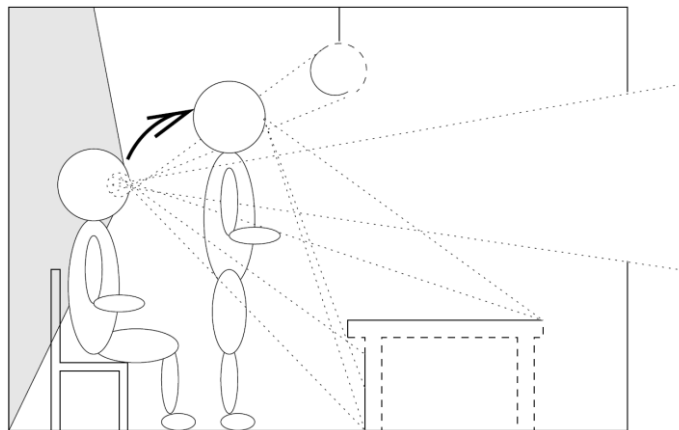
### **2. Informations structurées**

L'environnement est un terme qui réfère aux objets qui font partie du quotidien des êtres vivants. Il ne s'agit pas d'un espace physique, ni d'objets géométriques situés à des positions dans un monde à trois dimensions. Le terme "écologique" accolé à celui de "perception" vient du fait que les définitions des concepts mis en jeu, sont celles qui permettent à l'animal de vivre et de se comporter, et qui ont permis à l'animal de s'adapter au cours de son évolution. Par exemple, pour les êtres humains,



l'environnement est structurellement hiérarchisé en montagne, colline, bosse et rocher. De même, il n'existe pas de temps abstrait qui s'écoule de façon uniforme, mais des événements qui rythment la vie de tous les jours. Ces événements sont définis par les changements que les éléments de l'environnement subissent. Les événements sont eux-mêmes hiérarchisés et peuvent spécifier des saisons, le jour et la nuit ou le débit d'une rivière par l'observation de la vitesse d'écoulement.

L'être humain perçoit le monde par l'intermédiaire du flux optique ambiant (image suivante). Le flux optique ambiant est composé d'angles visuels formés par les surfaces visibles. Lorsque l'observateur se déplace, les informations de perspective changent, mais les invariants de structure ne changent pas.



### 3. Invariants

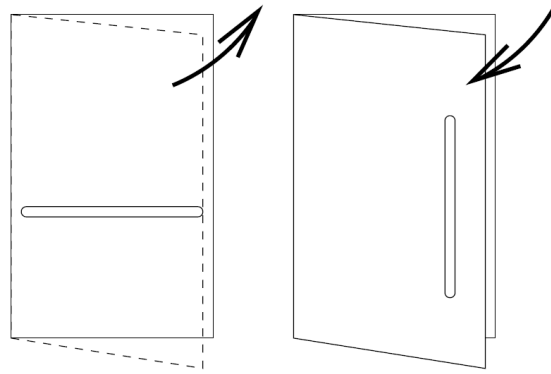
Pendant le processus de perception, il y a en parallèle une conscience de la persistance et une conscience du changement. Ainsi, à chaque déplacement, le flux optique change, tout en conservant certaines propriétés. Il n'y pas de changement soudain de structure : un objet de forme ronde ne passe pas brusquement à une forme carrée par exemple. Ces propriétés sont dites invariantes par rapport au déplacement, ou invariantes par rapport à la perspective. Le fait de se déplacer vise en partie à découvrir les invariants de perspective qui ne sont pas perceptibles si le flux ne change pas. Ce qui change constitue l'information qui spécifie le mouvement. Ce qui ne change pas spécifie la structure de la disposition des objets. Ainsi, une table est perçue en tant que telle lorsqu'on découvre qu'elle est constituée d'une planche horizontale soutenue par des pieds. C'est en modifiant la vue, en changeant le flux optique que l'on perçoit les relations qui lient les pieds à la planche, et que l'on découvre qu'ils la soutiennent. Bien sûr, il est possible de reconnaître une table sans faire de mouvement. En revanche, apprendre à connaître une table passe par le mouvement.

Les invariants concernent aussi les changements par rapport au temps qui passe : vieillissement des structures, dégradation des couleurs. Quelles que soient les entités impliquées, on retrouve les mêmes invariants.

### 4. Affordances

La perception commence par les surfaces apparentes des objets à percevoir. La question qui se pose est de savoir comment nous passons de la perception des surfaces à la perception de ce que l'environnement propose ou permet à un observateur. Gibson

définit les affordances d'un artefact comme les propriétés de cet objet qui peuvent être utiles à un animal. Selon lui, la perception de la composition et de la disposition des surfaces conduit à la perception de leurs affordances, et il y a perception directe des propriétés d'un objet relative à un animal. Dans l'image suivante, la porte de gauche a une poignée horizontale et basse, elle doit être poussée. La porte de droite a une poignée verticale, facilement préhensible et loin des gonds : elle "suscite" le fait d'être tirée [Norman, 1995], elle est « affordante ».



Pour Gibson, les sensations ne sont pas associées à des significations par un processus mental. De même, les objets ne sont pas décorés d'affordances par un processus mental. Les affordances sont perçues directement, et elles le sont car elles sont spécifiées dans les informations de stimulus. Nous avons vu que nous percevons les invariants de structure. Les affordances peuvent être considérées comme des invariants d'invariants. Par exemple, c'est la combinaison des invariants de la perception proprioceptive (la perception de son corps) et des invariants d'un objet qui permet d'évaluer les propriétés de cet objet relativement à l'observateur. En percevant à la fois un objet et lui-même, par exemple ses mains, l'observateur peut évaluer ce qu'il peut faire de l'objet, par exemple le fait d'être saisi par une main. Nous ne percevons le monde que parce que nous savons ce qu'il y a à percevoir.

## 5. Théorie de l'acquisition d'informations

En expliquant sa théorie de l'acquisition d'informations, Gibson redéfinit les principaux concepts impliqués dans la perception. Ainsi, la perception est vue comme une activité continue, sans pause, qui nous sert à rester en contact constant avec le monde. Les informations de spécification des objets dont se sert le processus de perception sont directement disponibles dans le flux optique. L'acquisition d'information se fait à l'aide de systèmes perceptuels, dont le comportement peut être décrit selon une boucle perception-action. Ces systèmes perceptuels s'affinent et s'adaptent, optimisent leurs ajustements. L'observateur distingue ce qui change et ce qui ne change pas : il perçoit les invariants du flot de sensations et ce qui varie avec le flot de sensations. Le processus d'acquisition ne se sert pas de la mémoire pour fonctionner. L'information n'a pas à être gardée en mémoire, car elle est toujours disponible.

## 6. Conclusion et application dans le domaine des visualisations

Dans ce paragraphe, nous donnons notre transposition de la perception écologique dans le domaine des visualisations (ou des Interfaces Homme Machine).

Les visualisations qui utilisent la perception écologique nécessitent peu d'apprentissage pour être interprétées (elles utilisent les connaissances de l'Homme dans son environnement naturel). La charge cognitive pour la perception d'information s'en trouve réduite, ce qui est un avantage important dans le domaine du contrôle aérien et des interfaces critiques.

Les visualisations écologiques sont plus efficaces pour transmettre de l'information (faible charge cognitive), mais il n'existe pas de méthode systématique permettant aux concepteurs de réaliser des designs écologiques. La caractérisation et l'évaluation des visualisations écologiques sera étudié dans le chapitre « design écologique et émergence ».

### III. Analyse des visualisations

Dans ce chapitre, nous nous intéressons à l'analyse des visualisations du Contrôle Aérien avec les outils disponibles en InfoVis. L'étude de visualisations spécifiques d'un domaine permet aussi sa généralisation en appliquant des résultats à l'ensemble des visualisations du domaine de l'InfoVis.

Cette analyse permet la réalisation d'une description qui est composée de deux éléments distincts : le *contenu* intrinsèque de la description et la *mise en forme* de la description. La mise en forme permet de présenter cette description et ainsi favorise la comparaison.

#### A. Le contenu

Le contenu de la description consiste à lister de manière exhaustive toutes les informations importantes permettant de décrire la visualisation (avec le type des données affichées et les éléments graphiques qui la composent).

##### 1. La caractérisation par le typage des données

Ce chapitre est consacré à l'explication des types de données et à leur utilisation. L'analyse « data scale » [Stevens, 1946] et par la suite la sémiologie graphique [Bertin, 1983] sont les premiers outils de description par le type des données dédiés à la caractérisation des visualisations. Le tableau suivant donne les différentes dénominations les plus couramment rencontrées et les auteurs qui les ont introduites :

<b>Stevens</b> [Stevens, 1946]	<b>Ware</b> [Ware, 2000]	<b>Bertin</b> [Bertin, 1983]
Nominal	Catégorie	Qualitative
Ordinal	Entier	Ordonnée
Intervalle	Nombre réel	Quantitative (métrique)
Ratio		

##### a) La théorie des échelles de Stevens : une taxonomie des échelles des nombres

Dans la théorie des échelles (« Theory of Scales Measurement »), toute donnée peut être catégorisée par une des échelles perceptives [Stevens, 1946] : *Nominal*, *Ordinal*, *Intervalle*, *Ratio*. Ces échelles font référence aux propriétés intrinsèques des données :

- les données N « Nominales » ont la propriété d'être différenciées des autres données (l'indicatif d'un avion est une donnée Nominale),
- le type O « Ordinal » permet à une donnée d'être perçue avec une magnitude différente et ainsi permet d'être qualifiée de supérieure, inférieure ou égale à une autre donnée (le niveau d'alarme d'un avion est une donnée Ordinale),
- les « Intervalles » I contiennent les mêmes propriétés que pour le type Nominal et Ordinal mais ajoutent une notion de régularité dans la magnitude perçue (le temps est une donnée Intervalle car il a une vitesse de défilement constante. Des

intervalles de temps de même durée seront perçus de la même façon). De plus l'intervalle de valeur nulle n'a pas de sens,

- les « Ratios » R ont toutes les propriétés des types Nominal, Ordinal et Intervalle, et ils ajoutent l'utilisation de la valeur nulle. Ainsi la comparaison de données de type Ratios est plus complète et permet l'utilisation d'opérateurs mathématiques. Par exemple on peut dire qu'une quantité vaut deux fois une autre valeur.

## b) Le type des données de Ware

Le but d'une visualisation, d'après Ware [Ware, 2000], est de transformer des données en formes visuelles de manière à optimiser leur perception par les utilisateurs. Pour ce faire on peut utiliser le type des données à afficher pour choisir le type de représentation le plus adapté.

Comme Bertin, Ware définit deux formes fondamentales pour les données :

- la *valeur* des données (Data Value), aussi appelée *entité* qui est la représentation visuelle de la donnée,
- la *structure* des données, aussi appelé *relation* qui définit les relations entre les entités. Ces relations sont parfois données explicitement (dans le cas de l'utilisation de la visualisation comme un outil pour mémoriser ou communiquer), et parfois ces relations sont à découvrir (dans le cas d'une visualisation pour le traitement et l'exploration des données).

Chacune des entités et des relations utilisent des *attributs* pour être visualisées. On peut comprendre ce terme comme étant les paramètres qui servent à décrire une entité. Par exemple, la couleur d'une pomme est un attribut, la température de l'eau aussi. Ce sont ces attributs qui sont qualifiables avec une échelle de données (par exemple celle de Steven [Stevens, 1946]).

Parmi les échelles définies par Stevens, Ware en a isolé trois qui sont majoritairement utilisées dans le domaine informatique :

- *catégorie* : qui correspond au type nominal N de Steven,
- *entier* : qui correspond au type ordonné O des éléments discrets,
- *nombre-réel* : qui combine les propriétés intervalle I et ratio R de Steven.

## c) Le type des données de Bertin

Selon Bertin, il existe trois niveaux d'organisation des données que l'on cherche à afficher (aussi appelé composante du plan) : *qualitative*, *ordonnée* ou *quantitative*.

Le niveau *qualitatif* (aussi appelé *nominal*) regroupe tous les concepts de simple différenciation. Une composante est qualitative lorsque ses catégories ne s'ordonnent pas de manière universelle. Elles sont en conséquence ré-ordonnables de différentes manières. Bertin précise que certaines composantes ont un ordonnancement qui dépend de l'activité ou de la culture, qui n'est donc pas universel. Bertin donne l'exemple du commerce portuaire, où les catégories [charbon, pétrole, blé, laine, coton, vin] sont ordonnées naturellement par leur masse volumique alors que ces catégories sont

qualitatives. La notion de distance entre catégorie n'a pas de sens : les données de type qualitatif sont équidistantes les unes par rapport aux autres. Leur transcription graphique ne doit donc pas faire apparaître à priori de regroupement (position des objets dans le plan) entre les catégories nominales.

Le niveau *ordonné* groupe toutes les catégories qui sont susceptibles d'être organisées de manière universelle (i.e. ordonnancement partagé par une large majorité de personnes) : ceci est plus que cela, et moins que cet autre. Bertin définit un ensemble de thèmes de catégories ordonnées :

- temporelles : âge, génération, état matrimonial, généalogie...
- appréciation de sensibilité : chaleur, vision des nuances de gris, taille des objets, distance des objets, santé...,
- habitudes sociales : hiérarchie sociale, militaire, administrative.

Les catégories ordonnées sont, comme les catégories qualitatives, équidistantes les unes aux autres.

Le niveau *Quantitatif* (ou *métrique*) permet à « une série de nombres de préciser la variation de distance entre les catégories ». La distance entre les catégories a un sens et ainsi il est possible de créer des groupes. Bertin ajoute une différenciation lorsque des types quantitatifs sont utilisés pour représenter un nombre d'objets comptés (taille de la population, nombre d'immigrants...). Il faut alors prendre en compte les variables dépendantes aux nombres comptés : par exemple la taille de la commune pour la population, les classes d'âges pour le nombre d'émigrants. Si on omet les variables dépendantes, les comparaisons entre catégories quantitatives n'ont pas de sens.

## **2. Exemple d'analyse de visualisation avec le type de données**

Dans cette partie, nous allons illustrer l'utilisation du type de données pour comparer des visualisations. Le premier exemple explicite les types des données affichées par l'écran radar des Contrôleurs Aériens. Le deuxième exemple (le papier Strip qui contient toutes les informations de vol d'un avion) montre comment le type des données permet de différencier les visualisations les unes des autres.

### **a) L'écran radar des Contrôleurs Aériens**

Pour assurer les services du Contrôle Aérien, plusieurs types de visualisations radar sont utilisées. Elles affichent la position des avions et permettent de répondre à des tâches précises : surveiller les séparations spatiales entre les avions, prévoir leurs évolutions et détecter les futurs conflits potentiels (passage sous les normes de séparation), diriger les avions sur leur route prévue, assurer la fluidité du trafic (minimiser les retards).

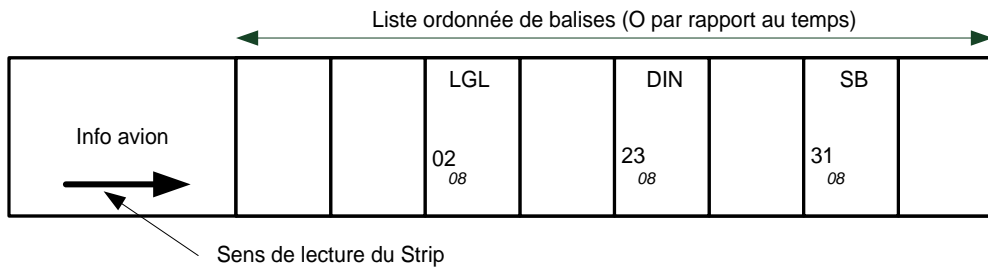
Pour répondre à toutes ces tâches, des données sont affichées avec des types différents. Par exemple le nom de l'avion est une donnée Nominale (N), le niveau d'importance des vols est une donnée Ordonnée (O), la distance entre avions est une donnée Quantitative (Q) et leur visualisation permet de répondre à ces questions :

- « Comment peut-on identifier un avion ? » Cet avion AF1234 est différent de celui-ci AF7890. Le type nominal (N) est alors utilisé.



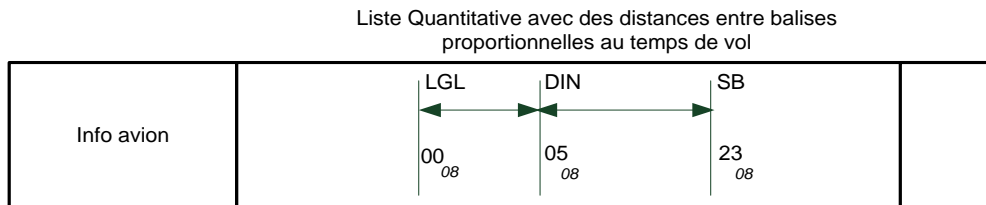
### (1) Les strips avant 1972

Les strips ont subi plusieurs évolutions. Par exemple avant 1972 les routes aériennes n'étaient pas nombreuses, chaque case était liée à une balise : la partie concernant le vol de l'avion était divisée en 10 cases. Le nom de la balise était effectivement imprimé si la route de l'avion devait passer par cette balise, sinon la case était vide. Cette association case/balise permettait aux Contrôleurs Aériens de comparer visuellement les estimées des avions aux mêmes balises. En revanche, l'association case/balise ne permettait pas de différencier deux avions qui suivent la même route mais en sens opposé. Pour palier ce problème, une flèche était ajoutée pour indiquer le sens de cheminement.



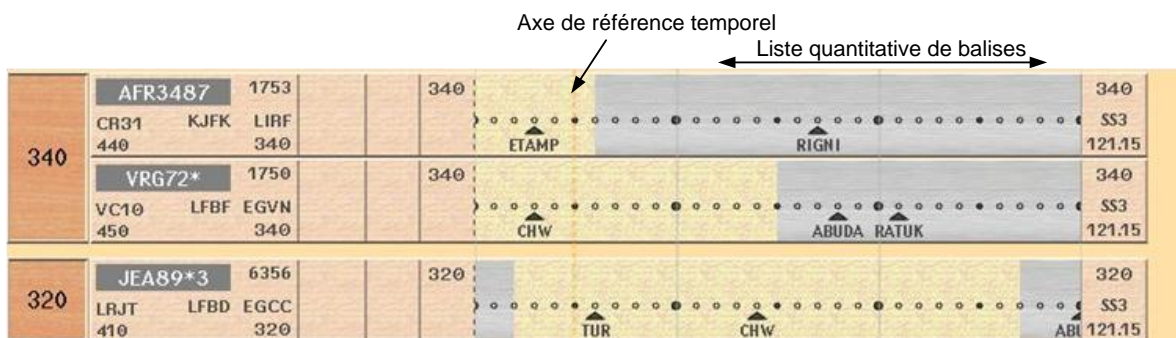
### (2) Base temps, une étude de strip Quantitatif

Alors que les Strips traditionnels affichent les balises de référence de façon ordonnée (O par rapport au temps), les chercheurs du CENA (Centre d' Etude de la Navigation Aérienne) [Nobel & Sperandio, 1973] ont étudié une version avec le type Quantitatif (Q) pour afficher la route de l'avion. Ainsi, la distance qui sépare les balises informe les contrôleurs du temps de parcours de l'avion. La lecture des strips se fait de gauche à droite avec un empan de 70 minutes.



### (3) Les strips animés

A l'instar de « Base temps », de nouveaux prototypes Q ont été réalisés en informatisant le Strip papier [Hering, 2004]. Ainsi, les strips sont dynamiques et utilisent l'animation avec un axe de référence pour indiquer l'heure courante.





### c) Comparaison par le type de données des différents designs des Strips

Tous ces différents designs de Strip essaient de répondre à la question générale de sécurité des aéronefs : « quels sont les conflits futurs ? », mais ils offrent des réponses efficaces à des questions différentes :

- « Quelle est la prochaine balise sur la route de l'avion ? » : il faut afficher une liste ordonnée O.
- « Quelle est la section la plus longue sur la route de l'avion ? » : il faut une liste avec des données quantitatives Q.
- « Où se trouve l'avion ? » : il faut une liste avec des données quantitatives Q et un axe de référence temporel pour indiquer l'heure courante.

### **3. La caractérisation par les éléments graphiques qui composent la visualisation**

La description par le typage des données n'est pas suffisante pour décrire une visualisation. Il faut aussi expliciter les éléments graphiques qui la composent. Tous ces éléments ont été étudiés dans le chapitre consacré à la perception. Nous pensons que cette description doit comporter les éléments suivants :

- la description de la fonction de transformation des données vers leur représentation visuelle,
- la liste des Variables Visuelles utilisées dans la visualisation,
- les informations de haut niveau comme la perception liées à la Gestalt, à la perception pré-attentive ou encore à la perception écologique.

Cette liste n'est pas exhaustive, elle soulève le problème de la recherche des dimensions capables de caractériser les visualisations.

## B. La mise en forme

Dans ce sous-chapitre, nous détaillons les travaux précédents consacrés à la mise en forme des informations issues de la caractérisation de visualisations. Ces travaux utilisent des structures graphiques différentes pour afficher la description des visualisations, c'est pourquoi nous les appelons *la mise en forme*. Le but est de permettre à l'utilisateur de mieux comprendre les informations disponibles dans les visualisations et ainsi de permettre leur comparaison. Les outils visuels sont de plusieurs formes : des tableaux, du texte, des graphiques.

### 1. La caractérisation de Card et Mackinlay

Card et Mackinlay (C&M) [Card & Mackinlay 1997] ont créé un modèle pour la description et l'ont utilisé pour décrire 12 visualisations (visualisations scientifiques, graphes de données multidimensionnelles, réseaux, arbres...). Ce modèle est une nomenclature. Il utilise une représentation sous forme de tableau qui permet de résumer les informations principales contenues dans chaque visualisation. Card et Mackinlay indiquent que leur travail va permettre à l'utilisateur d'analyser les portions de l'espace de design ainsi décrit et par extension de suggérer de nouveaux visuels. Malheureusement, ils n'indiquent pas comment utiliser concrètement leur description.

#### a) Le tableau de C&M

Le modèle de caractérisation se base sur une représentation sous forme de tableau :

Données				Perception automatique				Perception contrôlée			
Nom	D	F	D'	X	Y	Z	T	R	-	[]	CP

Chaque ligne du tableau correspond à une donnée affichée. La colonne D indique le type de la donnée (Nominale, Ordonnée, Quantitative). F est une fonction ou un filtre qui transforme ou qui crée le sous ensemble D'. A l'instar de D, D' est typée (N,O,Q). Les colonnes X, Y, R sont dérivées des variables visuelles de Bertin [Bertin, 1983]. L'image a trois dimensions : X, Y, Z plus le temps T. R correspond à la perception rétinienne de Bertin qui explicite la méthode employée pour représenter visuellement l'information (couleur, forme, taille...). Les liens entre les entités graphiques sont notés avec -, et la notion d'encapsulation est symbolisée par []. Enfin une distinction est faite si la représentation de la donnée est traitée par notre système perceptif de manière automatique (avec une faible charge cognitive) ou contrôlée (lecture de texte donc forte charge cognitive).

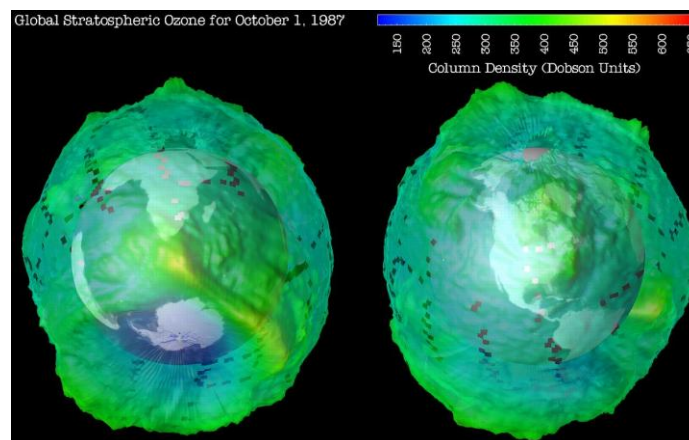
Ce tableau est rempli à l'aide des symboles suivants :

Symboles	Signification	
D	Type de données	
	N	Nominale
	O	Ordonné
	Q	Quantitative
	Qx	Quantitative et spatiale
	QXlon	Géographique
NxN	Nominale sur deux axes	
F	Filtre ou fonction pour modifier les données	
	F	Fonction non spécifiée
	Fs	Fonction de ré-ordonnement
	Mds	Mise à l'échelle multi dimensionnelle
	>	Filtre avec rangeslider ou menu
	Sl	Slider
D'	Type des données modifiées	
X, Y, Z, T	Position spatio-temporelle	
*	Sémantique autre que spatio-temporelle	
R	Propriété rétinienne (couleur, taille, forme, luminosité)	
-	Connexion	
[]	Contenant	
CP	Perception contrôlée (texte)	
P, L, A, S, V	Type de design : P (point), L (ligne), A (zone), S (surface), V (volume)	

### b) Exemple : visualisation scientifique

Les visualisations scientifiques utilisent des données intrinsèquement spatiales. C&M utilisent une notation spécifique pour le type de données géographiques avec Qx et Qy pour les données quantitatives intrinsèquement spatiales et Qlat et Qlon pour les données quantitatives de coordonnées géographiques. Les visualisations scientifiques utilisent aussi la même correspondance entre l'écran et les données : Qx-> X écran et Qy-> Y écran.

C&M ont, par exemple, caractérisé l'animation de la couche d'ozone [Treinish, 1994].



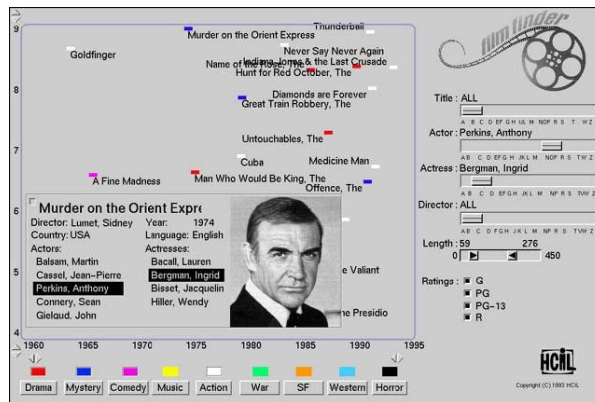
Dans le cas de l'animation de la couche d'ozone, une troisième dimension est utilisée pour coder la profondeur : Qz-> Z écran. Enfin, la couleur est utilisée pour coder la

densité d'ozone. Le tableau suivant est la caractérisation produite par C&M pour décrire la visualisation de la couche d'ozone :

Nom	D	F	D'	X	Y	Z	T	R	-	[]	CP
Lon	Qxlon	f	Qxlon	P							
Lat	Qylat	f	Qylat		P						
altitude	Qz	f	Qz			P					
Ozone	Q	F	Q					C			

c) Exemple : données multidimensionnelles

Dans le cas des visualisations de données multidimensionnelles, les dimensions de l'écran (X, Y) n'affichent pas obligatoirement des données intrinsèquement spatiales. Certaines dimensions des données peuvent être filtrées par des sliders. Dans FilmFinder [Ahlberg & Schneiderman, 1994], les sliders sont des filtres juxtaposés à la visualisation. L'utilisateur peut filtrer les films par une plage sur les noms, sur les acteurs, sur le style de film (drame, comédie...), la durée du film...



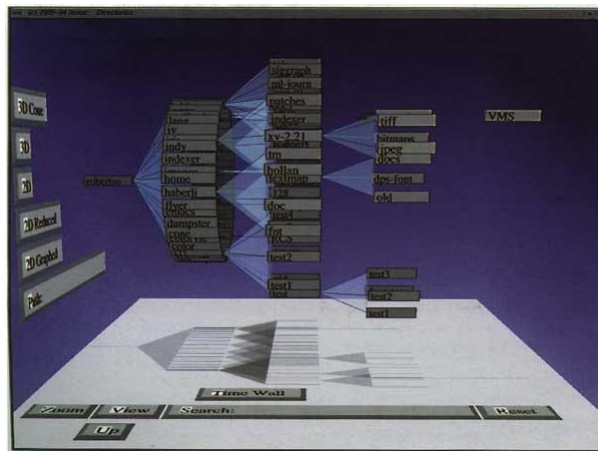
La visualisation permet d'afficher un graphique à deux dimensions des films sélectionnés avec en abscisse l'année et en ordonnée la qualité (bon, moyen, mauvais) du film. De plus le type de film est affiché avec un code couleur. Le tableau suivant est la caractérisation produite par C&M pour décrire FilmFinder :

Nom	D	F	D'	X	Y	Z	T	R	-	[]	CP
Année	Q	>	Q	P							
Qualité	Q	>	Q		P						
Type	N	>	N					C			
Titre	O	sl>									
Acteur	O	sl>									
Actrices	O	sl>									
Directeur	O	sl>									
Longueur	Q	br >									
Qualité	N	br >									

FilmFinder illustre le principe des « requêtes dynamiques » [Ahlberg et al., 1992]; la manipulation des sliders a un impact immédiat sur la visualisation avec l’affichage ou le non-affichage de marques. C’est pourquoi C&M ont grisé une partie du tableau de la caractérisation de FilmFinder.

#### d) Exemple : arbres

C&M ont caractérisé un ensemble de visualisations qui utilisent les arbres. Un exemple de représentation d’arbre est le Cone Tree [Roberstion et al., 1991]. De manière générale, les arbres sont de plus en plus larges lorsqu’on les explore en profondeur (la fonction est exponentielle en fonction de la profondeur). La solution proposée par les Cone Tree est d’enrouler chaque niveau de profondeur autour d’un cercle et d’utiliser l’interaction et l’animation pour laisser l’utilisateur choisir la partie du cercle à afficher.



Par exemple, un Cone Tree permet d’afficher l’arborescence d’un disque dur avec les fichiers qu’il contient. Chaque fichier a un parent. La fonction tree permet de transformer une donnée en entrée (le nom du fichier et son parent : NxN) en position 3D (XYZ). Le nom du fichier est affiché avec du texte, le lien qui le connecte avec son parent est affiché avec une ligne. Le tableau suivant est la caractérisation réalisée par C&M du ConeTree :

Nom	D	F	D'	X	Y	Z	T	R	-	[]	CP
Fichiers	NxN	Tree	XY Z	*	P	*			L		texte

## 2. La description de L. Wilkinson

Wilkinson présente une approche théorique pour étudier une grande majorité des représentations graphiques que l’on trouve dans les journaux scientifiques, dans les outils des statisticiens et dans les systèmes de visualisation de données [Wilkinson, 1999]. Pour ce faire, il utilise un formalisme textuel pour décrire chacune des visualisations étudiées. Ce langage textuel spécialisé dans la création de visualisations peut servir à caractériser les visualisations ; il est interprétable par un être humain mais aussi par la librairie de visualisation ViZml [Wilkinson , 1999].

### a) La notation de Wilkinson

La réalisation d'une visualisation est décomposée en trois étapes : la *spécification*, l'*assemblage*, l'*affichage*.

La *spécification* du graphique est exprimée à l'aide de six descripteurs :

- DATA : l'ensemble des opérations sur le Dataset qui créent les données à afficher,
- TRANS : la transformation des variables (exemple ; le tri),
- SCALE : la transformation d'échelle (exemple logarithmique),
- COORD : le système de coordonnées utilisé (exemple polaire),
- ELEMENT : taches (points, lignes, formes...) et leurs attributs esthétiques (comme la couleur),
- GUIDE : une ou plusieurs aides visuels (axes, légendes...).

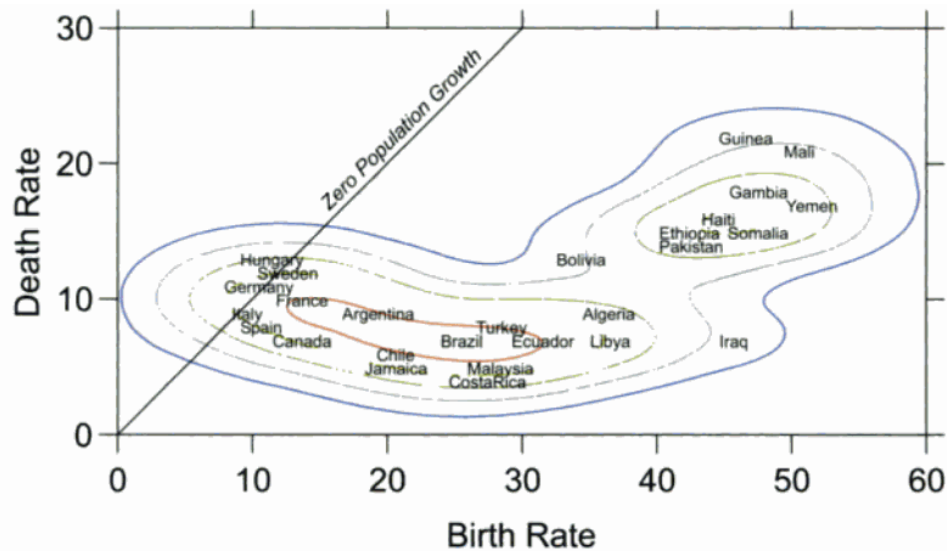
Le détail de cette notation et l'algèbre associé figurent dans un article de référence [Wilkinson, 1996].

L'*assemblage* s'attelle à décrire l'agencement des objets dans une scène : c'est le graphe de scène. Par exemple, l'assemblage est particulièrement important lorsque des objets dans la scène se chevauchent. L'assemblage permet alors de définir la structure de « profondeur » de la scène.

L'*affichage* définit la surface sur laquelle la visualisation est produite (papier, vidéo, hologramme...).

### b) Exemple de notation

Pour illustrer sa méthode d'analyse des visualisations, Wilkinson utilise, dans son ouvrage, un premier exemple de visualisation papier. C'est un graphe qui représente le taux de mortalité et le taux de naissance en 1990 pour 27 pays en utilisant une représentation scalaire (deux axes) et un noyau de densité (Kernel Density Estimation [Silverman, 1986]). Ce graphe permet de mettre en avant la démographie des pays et de les grouper.



La *spécification* de cette visualisation est la suivante :

```

ELEMENT : points(position(birth*death), size(0), label(country))
ELEMENT :
contour(position(smooth.density.kernel.epanechnikov.join(birth*death)),color.hue())
GUIDE : form.line(position((0,0),(30,30)),label(« Zero Population Growth »))
GUIDE : axis(dim(1),label(« Birth Rate »))
GUIDE : axis(dim(2),label(« Death Rate »))

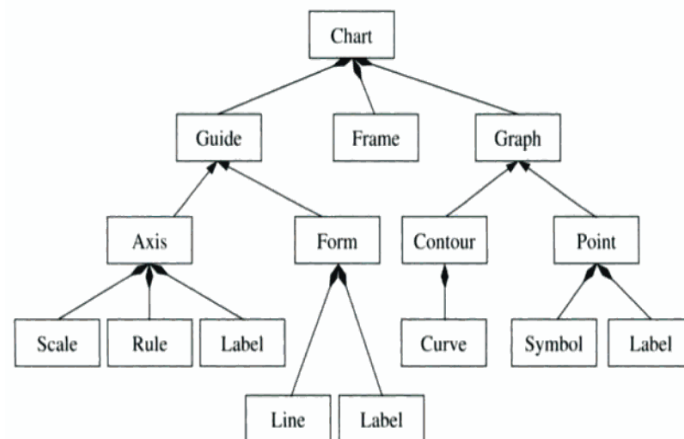
```

Cette *spécification* est composée uniquement d'ELEMENT et de GUIDE. Les ELEMENTs sont ici des points invisibles (taille zéro) avec un label (texte) pour indiquer le couple taux de mortalité/taux de naissance des pays. Les ELEMENTs contours affichent les transitions entre deux niveaux de densité dans le calcul d'une carte de densité (Kernel Density Estimate).

Les GUIDES permettent de donner des indications aux utilisateurs quant à la lecture du graphique. Les GUIDES décrivent les deux axes et une ligne de référence correspondant aux taux identiques entre le taux de mortalité et le taux de naissance.

Contrairement à Bertin, Wilkinson omet volontairement de décrire les guides dans les graphiques de son ouvrage. Ceci permet de simplifier leur description. Mais Bertin pense au contraire que les légendes et les aides à la lecture sont essentielles et ont une part importante dans la qualité des visualisations. On trouve ici un compromis entre exhaustivité (décrire complètement la visualisation) et compacité (garder un espace de description raisonnable).

L'assemblage décrit les agrégations (notions de groupes) et les héritages (notions de partage de mêmes propriétés) entre les objets. D'un point de vue informatique, pour recréer la visualisation il faudrait en plus de la description structurelle du graphique indiquer son ordre de lecture pour respecter les chevauchements entre les objets (avec par exemple l'utilisation de l'algorithme du peintre pour prendre en compte les chevauchements des objets). L'assemblage de la visualisation de la démographie des pays est la suivante :



Enfin *l'affichage*, qui est la dernière étape de Wilkinson pour décomposer une visualisation, est ici sans objet car le média (papier, écran ou autre) n'a pas d'importance pour la compréhension et l'utilisation de ce graphique.

### 3. La description de T. Baudel

Une partie des travaux de Baudel est consacrée à établir un modèle descriptif capable de représenter une classe importante d'algorithmes de visualisations d'informations avec un nombre restreint de paramètres [Baudel, 2002] [Baudel, 2007]. Baudel propose un langage capable de décrire une grande partie de l'espace de design des visualisations. Ce formalisme peut être utilisé pour mieux comprendre les caractéristiques de chaque visualisation.

Baudel décrit la classe de *visualisation compacte* comme étant la classe des visualisations qui se basent sur le modèle de Data Flow [Card & Mackinlay 1997]. Cette classe est équivalente à celle des représentations construites en temps linéaire : toutes les données sont disponibles et elles sont fournies sans discontinuité dans le modèle de transformation vers le visuel, le temps d'affichage est alors directement proportionnel au nombre d'objets à afficher. Les transformations des données vers le visuel sont les suivantes : partitions et sous partitions de l'espace des données, tri, génération de primitives avec des attributs graphiques. Le modèle de Data Flow est alors étendu pour inclure les partitions récursives et les variables locales.

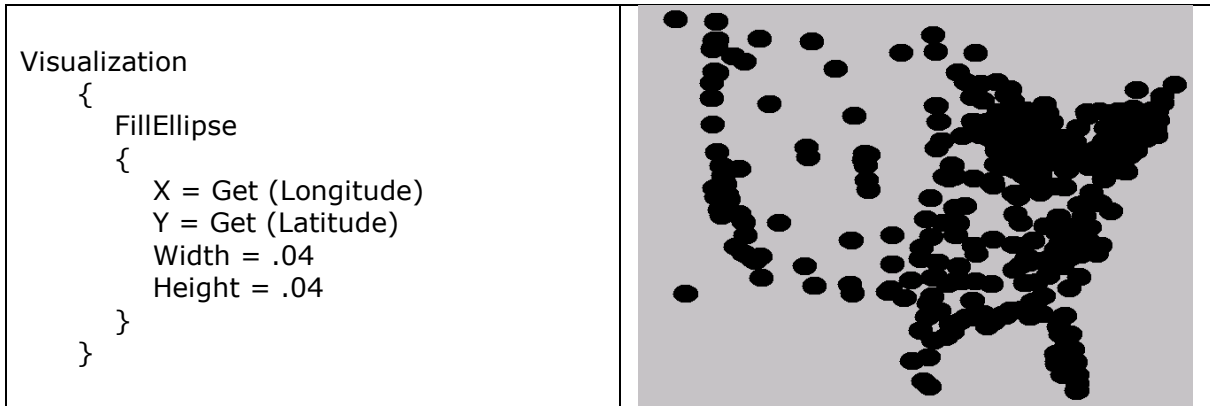
La description de Baudel est composée d'expressions, éventuellement conditionnelles ou itératives, qui peuvent comprendre des attributs de l'ensemble des données, des constantes ou des noms de variables locales. En ce sens, l'utilisateur ne fournit pas un paramétrage particulier d'un algorithme de visualisation mais une feuille de style décrivant les caractéristiques essentielles d'un algorithme de visualisation.

#### a) Exemples

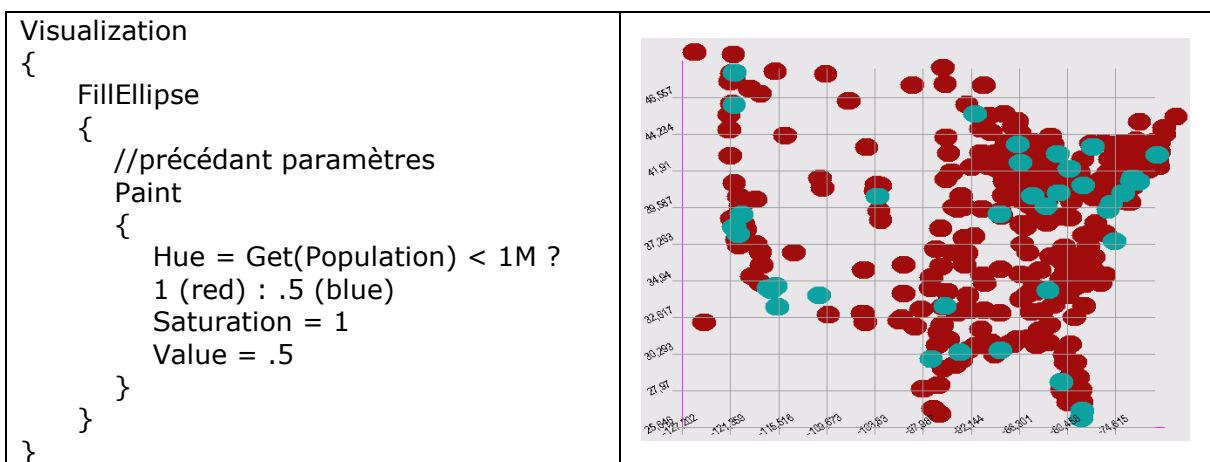
Dans ces exemples, les données sont des données statistiques sur les villes américaines avec leur population, leur taux de criminalité, le prix des maisons...

Dans la première partie, le graphe affiche une ellipse par enregistrement avec son positionnement proportionnel aux champs longitude et latitude.

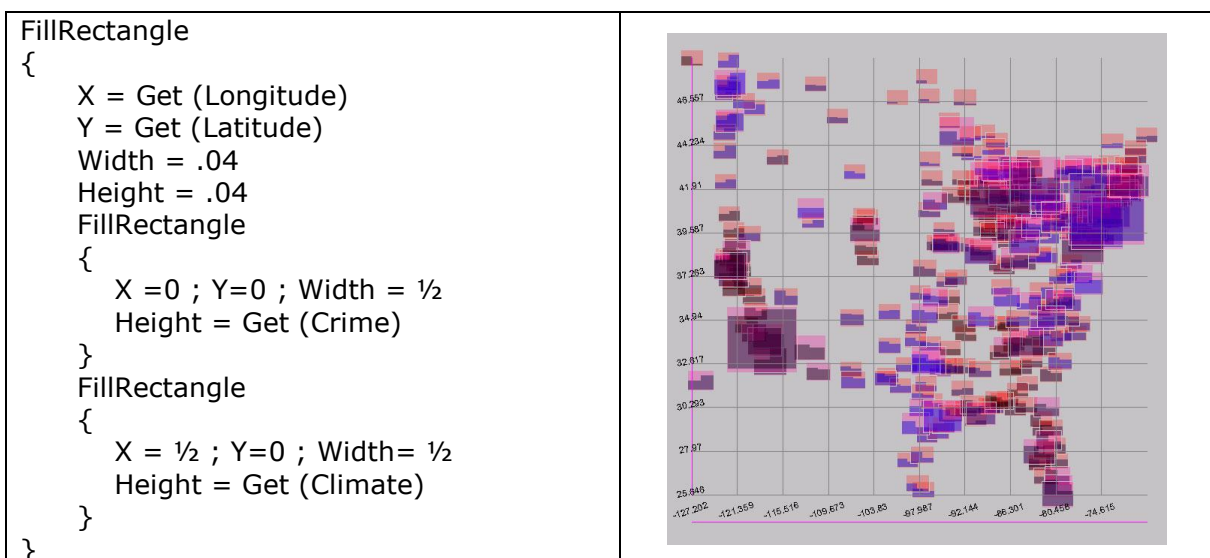




Il est possible d'ajouter des rendus conditionnels. Les ellipses sont rouges si la population est inférieure à 1 millions et bleus dans le cas contraire.



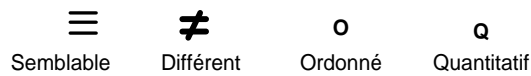
Il est aussi possible d'ajouter des éléments définis par le contexte de primitives. Par exemple, deux rectangles peuvent être dessinés à l'intérieur d'un rectangle père positionné sur chaque ville. Ces deux rectangles forment un *bar graph* avec les valeurs liées au climat et à la criminalité.



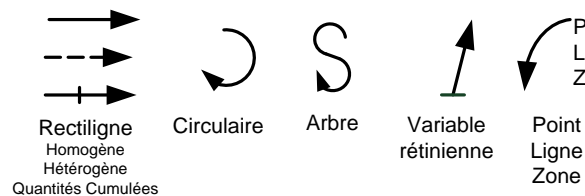
## 4. La description de J. Bertin

Jacques Bertin a travaillé sur l'analyse et la description des représentations graphiques [Bertin, 1983]. Il a mis en œuvre un système de codage visuel sous forme de signes et de symboles pour décrire les deux dimensions du plan. Il appelle cette caractérisation *l'imposition*. Il utilise une représentation symbolique des caractéristiques (appelé par Bertin des *niveaux*) des axes du plan (appelé par Bertin des *mobilisations des dimensions du plan*) qui définissent la construction graphique. D'après J. Bertin, ce codage permet d'analyser toutes les constructions imaginables et de fournir le schéma de construction du graphique.

Les niveaux d'organisation des composantes du plan sont codifiés visuellement à l'aide des signes :



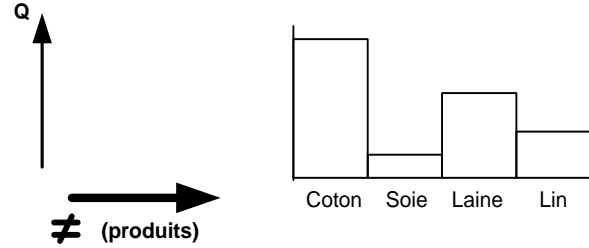
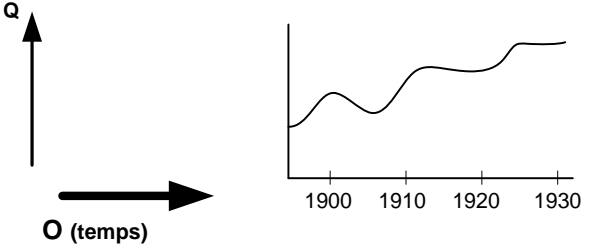
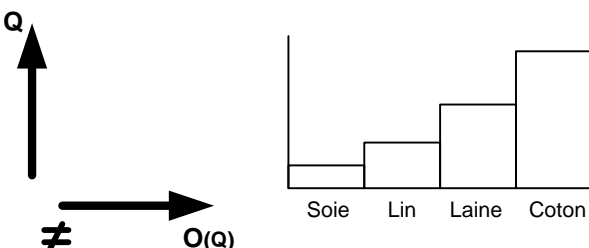
L'organisation des dimensions du plan, appelée par Bertin *la mobilisation des dimensions*, est caractérisée par les signes suivants :



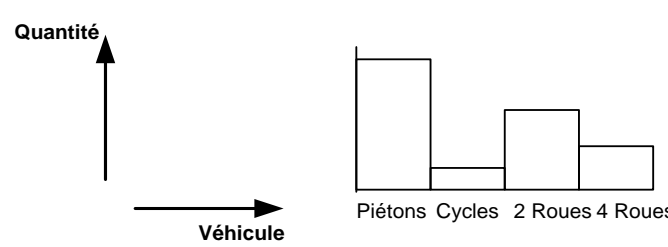
### a) Exemples de caractérisation

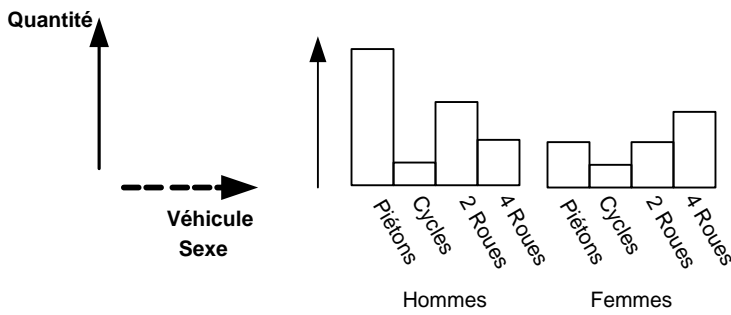
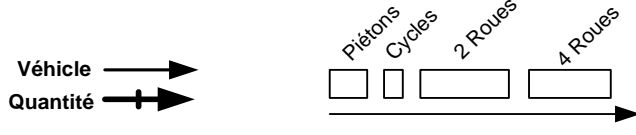

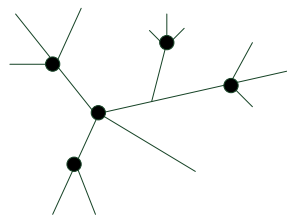
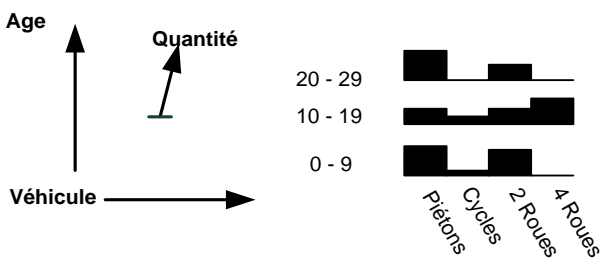
Les exemples suivants illustrent la caractérisation du niveau d'organisation des composantes du plan.

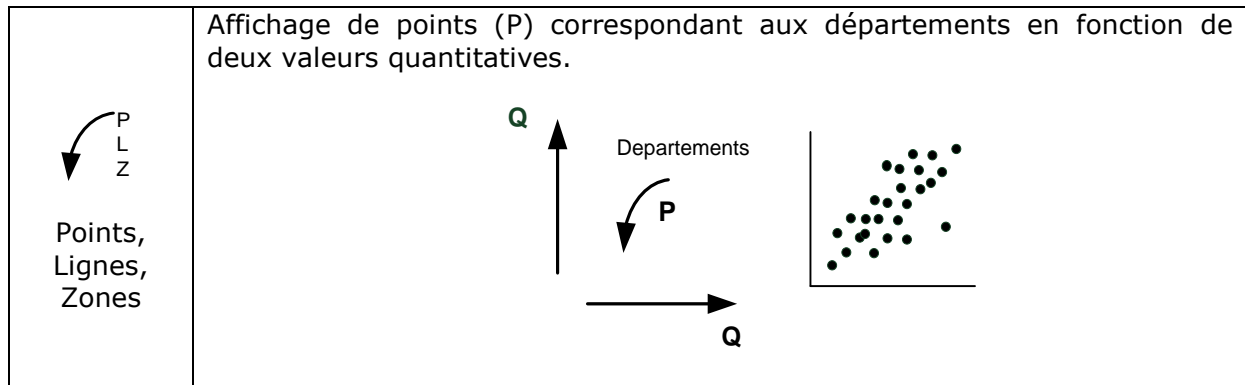
<p>≡ Semblable</p>	<p>Composante dont tous les éléments peuvent être vus comme <i>semblables</i>. Par exemple, les points utilisés pour représenter les départements dans la visualisation suivante sont semblables.</p> <div style="text-align: center;"> </div>
<p>≠ Différentiel</p>	<p>Sur l'axe des abscisses les produits sont affichés (composante Nominale N), sur l'axe des ordonnées la valeur affichée est quantitative.</p>

	
<p>○ Ordonné</p>	<p>Cet exemple est une série temporelle avec le temps (Ordonné O) en abscisse et une valeur quantitative en ordonnée.</p> 
<p>Q Quantitatif</p>	<p>Le dernier exemple représente des valeurs différentielles sur l'axe des abscisses qui sont ordonnées en fonction d'une valeur quantitative qui est affichée sur l'axe des ordonnées.</p> 

Les exemples suivants illustrent la caractérisation de la mobilisation des dimensions du plan.

<p>→ Homogène</p>	<p>La dimension du plan est homogène, elle affiche toujours la même dimension des données. De plus les catégories se suivent une fois pour toutes. L'axe des abscisses affiche le type de véhicule et l'axe des ordonnées affiche la quantité d'accident.</p> 
<p>---→ Hétérogène</p>	<p>La dimension du plan est hétérogène: les catégories se répètent plusieurs fois (notion de multiplexage). L'exemple affiche les mêmes informations que pour le graphe des accidents par type de véhicule mais avec une différenciation en fonction du sexe du</p>

	<p>conducteur.</p> 
<p>Quantités cumulées</p> 	<p>La dimension du plan affiche des quantités cumulées.</p>
<p>Circulaire</p> 	<p>Camembert des accidents par type de moyen de transport.</p>
<p>Arbre, réseau</p> 	<p>Arbre généalogique.</p>
<p>Variable rétinienne</p> 	<p>Une variable rétinienne est utilisée. Bertin utilise le terme « d'élévation au dessus du plan ». La visualisation utilise la troisième dimension du plan avec la taille pour coder la quantité de véhicules accidentés en fonction de leur catégorie et de l'âge du conducteur.</p>



## 5. Discussions

Dans ce chapitre, nous avons étudié trois types de mise en forme pour la caractérisation : le tableau, le texte, le signe graphique. Chacune de ces mises en forme a ses avantages et ses inconvénients, mais elles permettent toutes, à différents niveaux, la transmission synthétique des détails de la caractérisation de visualisations et leur comparaison.

### a) Discussion sur la caractérisation de C&M

Chaque tableau produit par C&M permet de décrire une seule fonction de transformation. Pour caractériser entièrement une visualisation il faudrait autant de tableaux que de types d'entités visuelles. C&M utilisent un seul tableau, car ils ne caractérisent que la *fonction principale* de transformation. Cette fonction principale est celle qui permet d'afficher l'information principale pour la tâche la plus courante de l'utilisateur de la visualisation : un fichier pour le Cone Tree, des films pour Film Finder, la qualité d'ozone...

C&M ont décrit un ensemble de visualisations sans en justifier le choix. De plus, ils ne fournissent pas de méthode ni d'exemple pour utiliser leur caractérisation. Néanmoins, Ils indiquent que leur description de l'espace de design permet de mieux le comprendre et de trouver par extension d'autres visuels non explorés.

Les travaux de C&M, même sans être parfaits, font parti des rares travaux à se consacrer à la description compacte (en terme de taille) et comparable (avec la comparaison des tableaux) de visualisations.

### b) Discussion sur la caractérisation de Wilkinson et Baudel

Les langages de description de Baudel et Wilkinson sont puissants car ils permettent de décrire une grande classe de visualisations. Ces langages sont compréhensibles par une machine ; leur interprétation permet la génération des visualisations décrites. L'utilisateur peut lui aussi lire ces descriptions et les interpréter, mais avec moins de précision que ne peut le faire un ordinateur. Ceci est particulièrement vrai avec les fonctions récursives qui utilisent les paramètres visuels d'entités graphiques ; l'utilisateur doit produire un effort important pour comprendre la caractérisation.

La comparaison de deux descriptions n'est pas simplement envisageable sans aide extérieure. Un utilisateur ne peut pas comparer directement deux descriptions

textuelles ; en revanche, à l'aide d'un logiciel de type « diff » la comparaison est possible en visualisant des différences entre chaînes de caractères.

### c) Discussion sur la caractérisation de J. Bertin

Bertin utilise un ensemble réduit de signes pour caractériser les dimensions du plan. Contrairement à la description de T. Baudel, la lecture de la description de Bertin ne permet pas de créer la visualisation. La description n'est pas suffisamment complète : il manque la description des échelles, la description des fonctions de transformation. De part la simplicité d'interprétation et la compacité de la description, la comparaison de deux descriptions est assez simple : l'utilisateur compare des représentations graphiques synthétiques.

## **6. Conclusion sur la mise en forme de la description de visualisations**

La description des visualisations permet de détailler les informations qu'elles contiennent, mais elle ne permet pas forcément de les exploiter. C&M disent vouloir utiliser leur mise en forme des descriptions avec des tableaux pour la réalisation d'une taxonomie de l'espace de design des visualisations sans pour autant donner des éléments concernant la marche à suivre. Selon nous, la mise en forme des descriptions permet principalement la comparaison, et ainsi permet de comprendre les différences entre les visualisations. Nous pensons donc que le contenu et la mise en forme sont deux éléments fondamentaux pour caractériser les visualisations. La séparation entre la mise en forme et le contenu est un des apports de cette thèse.

Notre démarche s'inscrit dans le long terme ; nous avons identifié un axe de recherche consacré à la comparaison des descriptions de visualisations. Nous pensons continuer dans ce domaine en proposant des descriptions sous forme de visualisations capable de décrire des visualisations : « une visualisation de visualisations ».

## IV. Evaluation des visualisations

---

Dans ce chapitre, nous nous intéressons à la qualité des visualisations. Ainsi nous avons recensé les travaux remarquables sur les choix de représentations pour la réalisation de visualisations efficaces. L'évaluation d'une visualisation peut être réalisée en vérifiant sa conformité avec des règles de design. Si une visualisation ne suit pas la règle, voire la transgresse, alors elle est mauvaise. Parmi les nombreuses règles de design, nous avons choisi de citer et d'illustrer celles de Bertin [Bertin, 1983], Tufte [Tufte, 1983], Wilkinson [Wilkinson, 1999] et Ware [Ware, 2000]. D'autres théories existent et sont toutes aussi valides mais moins connues comme celle de Chambers [Chambers, 1997] [Chambers et al., 1983], Cleveland [Cleveland, 1985] [Cleveland, 1995], ou encore les « design guidelines » de Kosslyn [Kosslyn, 1994].

### A. Evaluation par les règles de design

---

Une partie des règles de design de Bertin, Tufte, Ware ont été utilisées pour étudier et comparer la qualité de visualisations affichant des informations d'incertitude [Zuk, 2006]. Bertin, Tufte, Ware ont développé un certain nombre de principes pour réaliser des visualisations efficaces : Bertin se base sur la théorie des variables visuelles, Tufte utilise des principes de graphiques d'excellence, Ware s'appuie sur les principes de la perception et de la cognition.

#### 1. Bertin

Les règles de design définies par Bertin sont celles qui s'appliquent aux variables visuelles. Elles ont été décrites dans le chapitre sur la perception. Il a montré que les données affichées avec des variables visuelles qui correspondent à leur type (Associative, Sélective, Ordonnée, Quantitative) peuvent augmenter les performances cognitives pour certaines tâches [Bertin, 1983]. A ce titre, la variable visuelle *position* permet le meilleur support pour tous les types de données. Ne pas faire correspondre le type de données à afficher avec le type de Variable Visuelle ne permet pas de transmettre visuellement l'information sans erreur. Par exemple, le strip utilisé par les Contrôleurs Aériens affiche une liste ordonnée de balises (O) avec des espaces variables entre balises (position Q). Le type de donnée (O) et le type de Variable Visuelle utilisée (Q) ne correspondent pas, ce qui peut provoquer des ambiguïtés lors de la lecture des informations. A titre d'exemple, le strip radar n'utilise pas un visuel conforme : il affiche l'heure de passage des avions à chaque balise (Q) dans une liste ordonnée (O).

Bertin a identifié trois niveaux de perception d'information par la lecture d'une visualisation :

- le niveau *élémentaire* qui ne fait intervenir qu'un seul élément visuel,
- le niveau *moyen* qui fait intervenir des groupes d'éléments visuels,
- le niveau *supérieur* ou global qui concerne tous les éléments visuels.

Bertin précise qu'une visualisation qui n'offre que le niveau de lecture élémentaire est qualifiée de *figuration*, c'est-à-dire une illustration de type ornemental qui n'utilise pas les capacités perceptives humaines : c'est une visualisation de mauvaise qualité.

## 2. Tufte

Tufte a étudié de nombreuses visualisations et en a déduit des règles de design pour la création de visualisations efficaces nommées « visualisations d'excellence et d'intégrité » [Tufte, 1983]. Une visualisation est qualifiée d'excellence si :

- elle est correctement conçue avec une représentation de données pertinente pour une tâche spécifique, des données statistiques, et une structure visuelle,
- elle affiche des informations complexes et permet leur transmission avec clarté, précision, et efficacité,
- elle donne le maximum d'informations dans un temps réduit avec le minimum d'encre dans le minimum d'espace,
- elle affiche des données multidimensionnelles,
- elle affiche des données validées.

L'optimisation de l'encre pour le dessin est un principe qui oblige le designer à présenter le maximum d'information sur la plus petite surface et à éliminer l'encre superflue (i.e. n'apportant pas d'information). L'encre superflue peut distraire le lecteur par des analyses inutiles. La densité d'information fait référence au nombre de données affichées divisé par la taille de leur représentation. Si ce rapport est trop faible, le graphique doit être réduit en taille, ou les données doivent être reconsidérées ou être mises sous forme de table. Tufte indique qu'il est important de conserver le même design et de faire varier seulement les données pour assurer leur comparaison.

Pour éviter les visualisations trompeuses, il faut suivre des *règles d'intégrité*. Une visualisation est qualifiée d'intègre si elle respecte les règles suivantes :

- la représentation de valeurs numériques doit être directement proportionnelle à la valeur représentée sur la visualisation,
- le facteur de mensonge doit être proche de 1. Il est égal au rapport entre l'importance d'une valeur dans la visualisation et l'importance de cette donnée dans l'ensemble des données,
- la visualisation doit afficher les variations des données et non les variations de représentations,
- la visualisation ne doit pas utiliser de données hors de leur contexte,
- le nombre de dimensions utilisées pour la lecture des données ne doit pas excéder le nombre de dimensions des données qui sont représentées,
- les représentations monétaires doivent prendre en compte l'inflation de la monnaie utilisée,

Tufte a aussi introduit la notion de *design pattern*. Ainsi, comme en programmation, le designer peut utiliser une représentation « type » pour transmettre de l'information. Par exemple le design pattern appelé « petites miniatures » est utilisé pour comparer des données à l'aide d'un storyboard composé de petites images.



La règle de design la plus significative issue des travaux de Tufte est résumée dans le principe suivant: « un graphique d'excellence est celui qui donne à l'utilisateur le plus grand nombre d'informations dans le plus petit intervalle de temps avec le moins d'encre dans le plus petit espace ».

### 3. Ware

Ware a rédigé un ouvrage appelé Information Visualisation [Ware, 2000], qui traite de nombreux thèmes de recherche scientifique sur la perception et la compréhension des visualisations. En général, ces thèmes trouvent leur fondement dans les domaines de la physiologie et de la recherche cognitive, et moins dans les théories de Bertin et de Tufte. Les résultats de Ware sont dans la plupart des cas compatibles avec les travaux de Bertin et de Tufte mais avec des considérations physiologiques :

- visualisation pré-attentive : certaines variables visuelles sont qualifiées de pré-attentives [Treisman, 1985] (ce qui est à rapprocher de la sélectivité de Bertin). Par exemple la courbure, le groupement spatial, le flou, les amoncellements, le scintillement, la direction de déplacement, la profondeur stéréoscopique,
- comprendre grâce à la visualisation : une partie des travaux de Ware est consacrée à la discussion sur les articles traitant du sujet de résolution de problèmes par la visualisation. Il aborde les connaissances sur la mémoire iconique, la mémoire à long et court terme, et la mémoire visuelle. Les théories sur les motifs remarquables de parcours oculaire et leurs implications dans les procédures cognitives pour la résolution de problème sont abordées,
- les mots et les images : le texte est souvent supérieur à l'image pour présenter des idées abstraites, logiques, et des informations conditionnelles. Ware approuve les principes d'excellences des représentations de Tufte avec l'apposition de texte aidant la description. Il ajoute que l'association entre un élément graphique et son texte est rendue possible grâce à la propriété de proximité et de connexion des lois de la Gestalt [Kofka, 1935].

### 4. Wilkinson

Wilkinson [Wilkinson , 1999] souligne les difficultés rencontrées par les designers et les concepteurs de visualisations pour le choix des attributs esthétiques (ou visuels). Il est très difficile de prévoir comment les informations vont être perçues par les utilisateurs. Néanmoins, il a établi quelques règles de design :

- un attribut visuel doit être capable de représenter des données de type continu (Quantitative, Ratio) et de type catégorie (Intervalle, Nominal), mais un attribut n'est pas forcément linéairement perceptible,
- quand on représente une donnée continue, l'attribut doit varier principalement sur une dimension psychophysique. Pour utiliser des attributs multidimensionnels comme par exemple la couleur, on doit mettre à la même échelle ces dimensions pour les afficher, ou bien on doit les combiner à l'aide d'opérateurs mathématiques pour les fusionner,
- un utilisateur doit être capable de percevoir la valeur de l'attribut visuel distinctement et sans effort,

- un utilisateur doit être capable de percevoir la valeur de deux attributs dans un graphique. Cette tâche nécessite normalement, mais pas nécessairement, une attention sélective (l'utilisateur isole mentalement les éléments visuels à observer). La sélectivité n'est pas possible avec toutes les variables visuelles (Voir Bertin), et encore moins avec des paires d'attributs esthétiques. En tout cas, il ne faut pas utiliser d'attributs et de paires d'attributs qui ne soient pas clairement séparables (sélectifs). Kosslyn [Kosslyn, 1994] a proposé un ensemble de règles de design à appliquer pour les choix possibles d'attributs visuels,
- chaque attribut visuel doit correspondre à une donnée différente.

## **B. Autres méthodes d'évaluation**

---

L'application des règles de design permet d'accélérer le processus de design, mais la validation finale doit passer par une évaluation avec des utilisateurs. Néanmoins, il existe d'autres méthodes. Plaisant [Plaisant, 2004] a identifié le besoin d'outils pour évaluer les visualisations du domaine de l'InfoVis et donne des exemples d'évaluations. Ces évaluations sont réalisées avec des études longitudinales, des métriques, des dimensions de caractérisation. Par exemple, dans sa taxonomie, Tweedie [Tweedie, 1997] utilise deux dimensions de caractérisation. La première dimension caractérise les données en valeur et en structure. La deuxième concerne l'interaction avec les données : manipulation directe ou indirecte.

Selon nous, ces outils sont de trois catégories. Nous les avons classées par ordre d'efficacité :

- *a posteriori* : l'évaluation est effectuée après la réalisation de la visualisation,
- *prédictif* : l'évaluation est réalisée avant la production de la visualisation. Le designer peut prédire la qualité de sa visualisation,
- *prescriptifs* : ces outils donnent la solution de design en fonction d'un problème ou d'une tâche à réaliser. L'évaluation n'est pas utile.

L'évaluation par les règles de design est un outil *a posteriori*. Les taxonomies de visualisation sont aussi des outils *a posteriori* pour l'évaluation des visualisations. Les taxonomies permettent de comprendre les relations entre les visuels et dans une certaine mesure d'en tracer l'historique, de les classer en fonction de critères comparables, de découvrir des domaines non explorés, et de donner un catalogue utilisable pour la réalisation de futures visualisations. La méthode d'évaluation proposée par Wijk [Van Wijk, 2005] est un outil *a posteriori*. Cette méthode d'évaluation est basée sur des critères économiques comme le rapport entre utilité et utilisabilité des visualisations.

A notre connaissance, il existe très peu d'outils *prédictifs* dans le domaine de la visualisation. Des outils comme Keystroke [Card et al., 1980] ou encore CIS [Appert et al., 2004] permettent de prédire le temps que va mettre un utilisateur à réaliser une tâche. La transposition de ces outils dans le domaine de la visualisation permettra de prédire les informations qu'un utilisateur va potentiellement acquérir dans un certain contexte. Conversy a réalisé un outil capable de modéliser le chemin visuel pour répondre à une tâche [Conversy et al., 2010].

Selon nous, les seuls travaux se rapprochant des outils *prescriptifs* sont ceux de Casner [Casner, 1991] avec la production automatique de visualisations basées sur des besoins exprimés.

L'évaluation des visualisations est un domaine de recherche émergent ; un workshop y est consacré pour promouvoir des techniques d'évaluations innovantes non basées sur l'évaluation du temps et du nombre d'erreurs pour la réalisation d'une tâche « BEyond time and errors: novel evaLUation methods for Information Visualization » : BELIV 06, BELIV 08, BELIV 10.

## **V. L'exploration de données et le modèle de Data Flow**

---

Comme nous l'avons déjà expliqué dans l'introduction, une partie de cette thèse est consacrée à l'exploration de données. Pour être efficace, l'exploration de visuel suit une démarche indiquée par le mantra « Overview first, zoom and filter, then details on demand » de B. Shneiderman [Shneiderman, 1996]. D.Keim [Keim et al., 2006] a modifié cette démarche pour la spécialiser dans la fouille de données : « Analyse first, show the important, zoom, filter and analyse further, details on demand ».

Ce chapitre est divisé en trois, avec l'étude du modèle de DataFlow qui modélise le processus d'exploration de données, la visualisation des données, et finalement l'interaction avec les données.

### **A. Le modèle de Data Flow pour l'exploration de données**

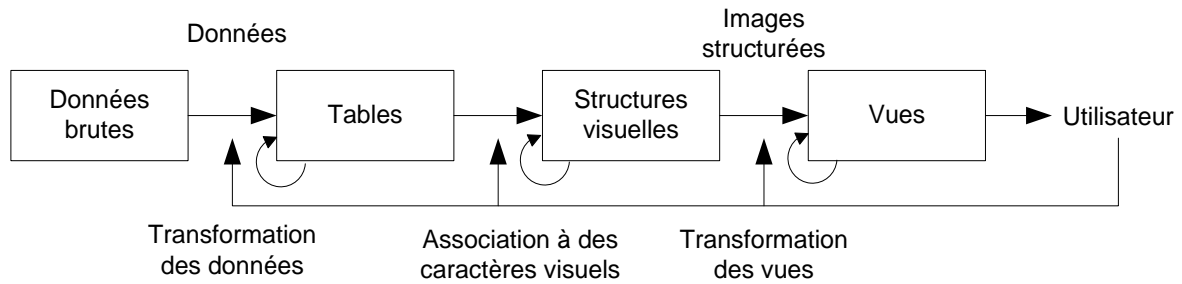
---

L'utilisation de visualisations pour la transmission d'informations est essentiellement liée au mécanisme mis en œuvre pour la représentation de données perceptibles par un utilisateur. Il existe des modèles de référence qui structurent ce mécanisme. Par exemple, le modèle de DataFlow décrit les principes qui permettent de faire la correspondance entre les données à afficher et leur représentation. Il permet aussi :

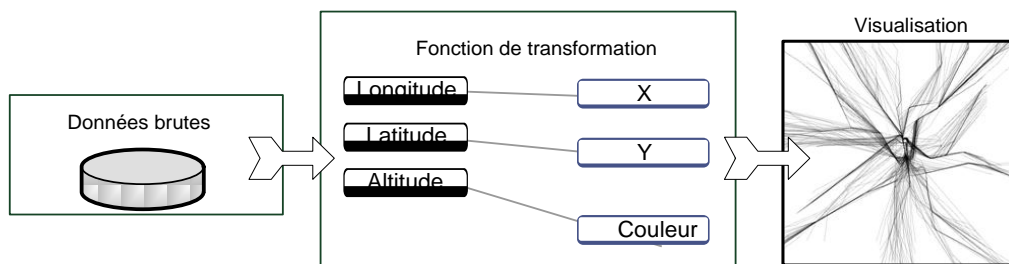
- de créer des descriptions abstraites des visualisations,
- d'offrir un cadre analytique qui permet une meilleure compréhension des techniques de visualisation,
- de décrire et d'identifier chaque processus du modèle,
- de séparer la partie visualisation de la partie interaction.

Dans ce chapitre, nous allons étudier le modèle de référence de Data Flow pour l'interaction et la transmission d'informations.

La visualisation d'informations est un processus impliquant le traitement des données depuis leur source jusqu'à leur visualisation. Le pipeline de visualisation d'informations est relativement long et complexe. Celui-ci a déjà fait l'objet de décompositions, comme le modèle introduit par P.K. Roberston [Robertson & De Ferrari, 1994] ou encore le « Data State Reference Model » introduit en 1999 par Ed Chi [Chi, 2000]. Nous allons décrire le modèle de Data Flow de Card, Mackinlay et Shneiderman [Card et al., 1999]. Ce modèle est présenté de façon linéaire avec en entrée les données à afficher, et en sortie leur visualisation. Ce modèle correspond parfaitement à une implémentation informatique et est suffisamment simple et complet pour produire une très grande classe de visualisations avec leurs interactions sur les données.



Dans ce modèle, les données sont transformées vers une représentation au travers de trois étapes paramétrables par l'utilisateur. La première transformation consiste à formater les données dans des tables. Ceci permet par exemple de rajouter des métadonnées (des données calculées à partir des données existantes), mais aussi de formater les données brutes dans des formats idiosyncratiques. Ensuite, ces tables sont utilisées pour faire la correspondance avec des éléments visuels (par exemple les variables visuelles de Bertin). Cette partie est le cœur de ce modèle, c'est le « mapping » des données vers le visuel ; c'est à ce niveau que la théorie des échelles [Stevens, 1946] est appliquée. Enfin, les entités visuelles ainsi définies sont transformées pour être affichées à l'écran. L'utilisateur peut, par exemple, interagir sur ce dernier étage en modifiant le zoom, les éléments à mettre au premier plan pour faire ressortir des patterns, des relations entre les données. Il n'est pas toujours nécessaire de transformer les données avec ces trois niveaux. Souvent un choix judicieux des dimensions pour les axes X et Y suffit à la réalisation de visualisations efficaces.



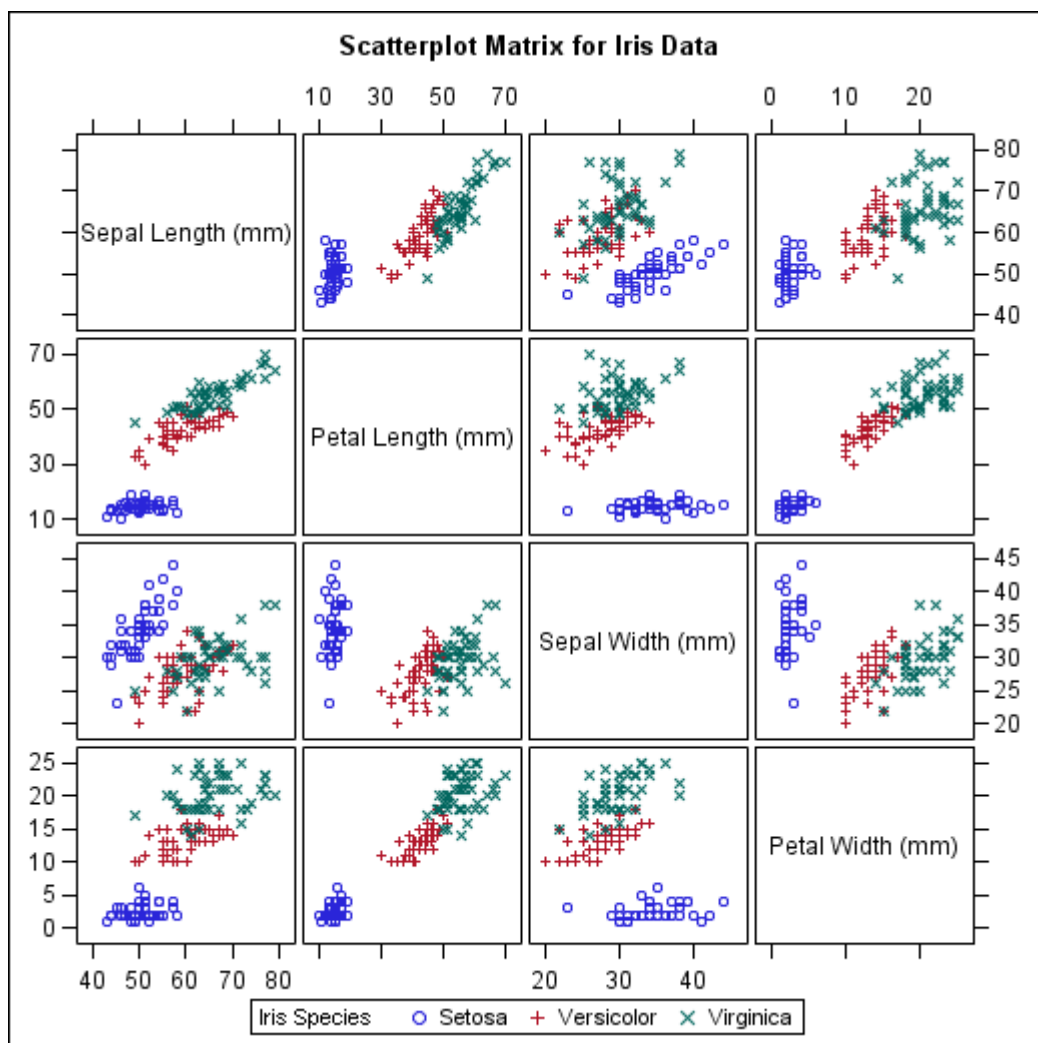
## B. La visualisation des données pour l'exploration

La plupart des logiciels d'exploration de données peuvent être modélisés avec le modèle de Data Flow [Card et al., 1999]. Ils sont capables d'afficher de grandes quantités d'informations à l'écran. Pour ce faire, ils utilisent différentes techniques d'affichage basées sur des vues multiples avec des systèmes de navigation différents. De plus, ils proposent différentes techniques d'interaction pour explorer les données. Les logiciels les plus connus dans le domaine de la recherche et d'exploration d'informations sont : Spotfire [Ahlberg, 1996], Polaris/Tableau [Stole & Hanrahan, 2000], GGobi [Swayne et al., 2003], TimeSearcher [Hochheiser & Shneiderman, 2004], SageTools [Roth et al., 1994]. Les toolkits pour l'InfoVis basés sur le modèle de Data Flow sont entre autres : InfoVis Toolkit [Fekete, 2004], ILOG Discovery [Baudel, 2004], VQE [Derthick et al., 1997], nVizN [Wilkinson, 1999].

## 1. Organisation de l'espace de travail et la navigation

Pour assurer la visualisation de données multidimensionnelles, il est possible d'utiliser des visualisations à deux dimensions (une dimension par axe). Pour afficher toutes les combinaisons d'axes, il faut afficher  $(n*(n-1))$  graphiques avec  $n$  dimensions disponibles dans les données. Il est alors possible d'afficher tous ces graphes à l'aide d'une matrice. Les visualisations de type scatterplots matrix [Cleveland & McGill, 1988] sont des vues positionnées et ordonnées de telle façon que toutes les vues respectivement sur une même ligne et sur une même colonne, affichent la même dimension des données.

L'image suivante montre les différentes combinaisons possibles pour l'affichage de graphiques avec comme données la longueur et la largeur des sépales et des pétales de 50 iris de quatre variétés différentes [Anderson, 1935]. On peut remarquer que la diagonale de la matrice qui fait correspondre sur l'axe des abscisses et des ordonnées la même dimension est utilisée pour afficher le nom de la dimension affichée, ce qui est une optimisation de l'espace disponible.

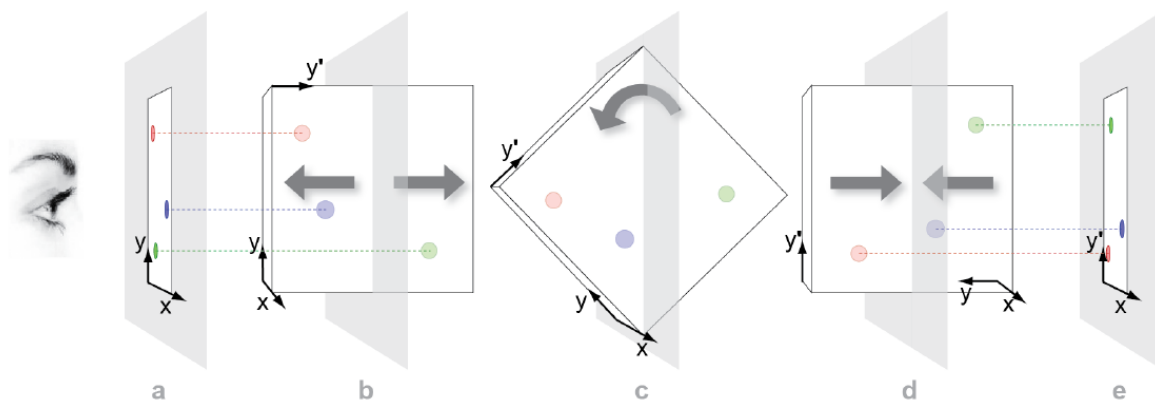


Toutes les combinaisons d'axes sont alors affichées. La matrice est symétrique et comporte de nombreuses redondances, mais elle permet de rapidement détecter les graphes qui sont intéressants à explorer : les graphes qui comportent des corrélations entre les dimensions, les tendances, et les groupes. La contrepartie de ce genre de

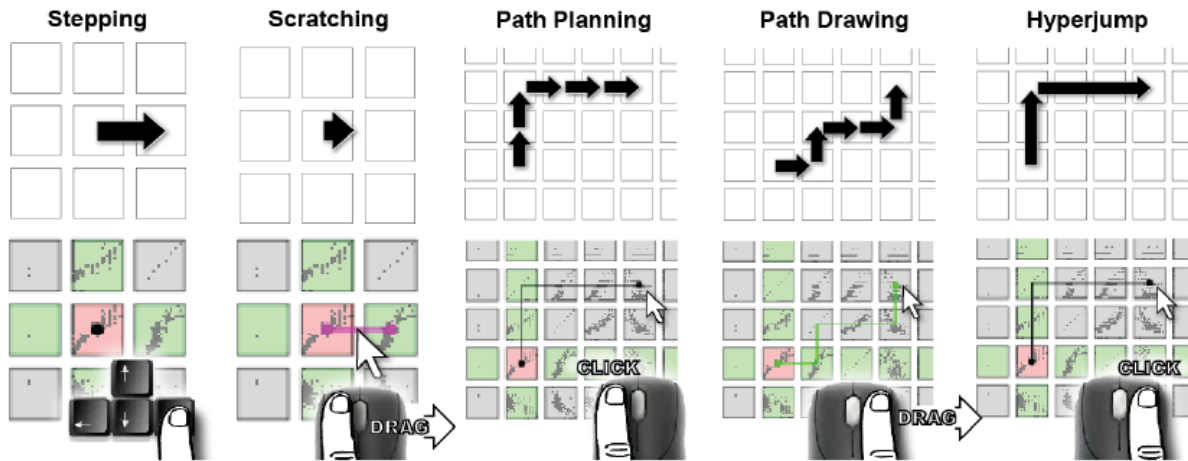
visualisation est sa taille qui devient rapidement un problème pour un grand nombre de dimensions à afficher. Il faut alors fournir à l'utilisateur des outils pour zoomer et naviguer dans la matrice, ce qui réduit l'intérêt de la vue d'ensemble des combinaisons des dimensions. Dans le logiciel Rolling the Dice, l'utilisateur a à sa disposition un ensemble d'outils de navigation sur la vue Scatterplot matrix [Elmqvist et al., 2008].

## 2. Transitions entre vues

Les vues de type Scatterplot Matrix donnent des informations sur la globalité des informations disponibles avec l'affichage de nombreuses miniatures. La taille du Scatterplot Matrix est problématique avec des données comportant de nombreuses dimensions : les vues sont réduites à des miniatures, et il devient difficile d'interagir avec elles (i.e. brushing). De plus, l'interaction sans la navigation oblige à chercher une vue particulière dans la matrice, ce qui peut prendre du temps, l'utilisateur doit trouver la ligne et la colonne correspondant aux deux dimensions qui l'intéressent. Dans le cas de la recherche d'informations précises, l'utilisateur doit utiliser une seule vue pour maximiser la surface d'affichage. Il faut alors lui fournir des outils pour modifier la configuration de sa vue, avec par exemple la modification des dimensions visualisées. Pour résoudre ce problème, les designers ont conçu des outils de Pan/Zoom qui permettent d'afficher une vue agrandie sur une partie de l'écran. Cette technique ne permet plus d'afficher correctement l'ensemble des vues de la matrice. « Rolling the Dice » offre un ensemble d'outils pour naviguer d'une vue à l'autre avec une animation sous forme de rotation [Elmqvist et al., 2008]. Rolling the dice n'affiche qu'une vue à la fois. Ce qui rend l'interaction entre deux vues plus longue car il faut alors basculer entre ces vues avec l'animation.



« Rolling the Dice » utilise plusieurs méthodes originales pour naviguer dans un scatterplot matrix.



En combinant l'utilisation d'une vue globale et de l'animation sous forme de rotation des axes du scatterplot, la navigation dans un jeu de données est rendue efficace.

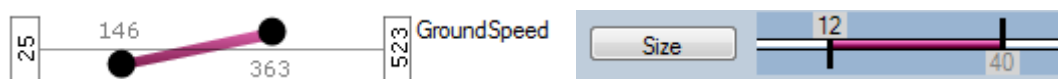
## C. Interaction avec les données

Pour rappel, Bertin a identifié trois utilisations des visualisations : communiquer, stocker ou comprendre des informations. Pour communiquer et stocker de l'information, la visualisation des données par un media est suffisante. Dans le cas de la compréhension et donc de l'exploration des données, l'utilisateur a besoin d'outils pour interagir avec les visualisations. Par exemple, l'utilisateur peut effectuer les actions suivantes : brushing (type de sélection), drill-down (recherche d'information), zoom... Toutes ces manipulations sont détaillées par Becker [Becker & Cleveland, 1987], Cleveland [Cleveland & McGill, 1988], Cook [Cook & Weisberg, 1994] et Swayne [Swayne et al., 2003].

Le modèle de Data Flow intègre les actions potentielles de l'utilisateur à chaque niveau de la transformation des données vers le visuel. Cette manipulation consiste principalement à effectuer des sélections en groupe de données. Ces groupes de plus petite taille peuvent ainsi être analysés séparément ou encore recombinaés avec d'autres pour faire ressortir des informations pertinentes.

### 1. Systèmes de filtrage et de sélection

Bien qu'originellement créées pour l'exploration de données, les requêtes dynamiques [Ahlberg et al., 1992] représentent un élément essentiel dans la formulation d'une requête. L'association du filtrage avec un rangeslider permet une manipulation rapide, immédiate et incrémentale d'un filtrage sur les données [Shneiderman, 1987]. C'est ainsi que le rangeslider réifie le principe de requête simple : « je veux les données comprises entre telle et telle valeur » en proposant un système de filtrage efficace.



D'autres systèmes proposent de sélectionner des données en définissant des zones ou des surfaces englobant des entités visuelles, qui changent alors d'apparence (par exemple les systèmes de lasso avec lesquels les objets inclus dans la zone de sélection changent de couleur). Dans les systèmes à vues multiples, comme par exemple les



Scatterplots Matrix, les données sélectionnées sont mises en évidence dans toutes les vues où elles sont présentes. C'est ainsi que l'utilisateur peut comprendre les relations entre les données au travers des différentes vues.

L'interaction pour sélectionner des données implique la désignation unitaire de chaque donnée [Livny et al., 1997], l'utilisation d'un rubber-band [Livny et al., 1997] [Elmqvist et al., 2008], l'utilisation d'un lasso [Livny et al., 1997], ou encore du brushing [Becker & Cleveland, 1987]. Plusieurs systèmes proposent des versions évoluées de brushing, comme XmdvTool [Ward, 1994]. Néanmoins, toutes ces techniques requièrent des paramétrages complexes et longs avant d'obtenir la sélection souhaitée, ce qui interfère dans le processus de requête rapide. Le problème réside le plus souvent dans l'utilisation du mode d'interaction : sélection, ou désélection. Les transitions d'un mode à l'autre prennent du temps (ouverture de menus par exemple) et nuisent ainsi à la dynamique de l'interaction.

Tous ces outils permettent de définir une sélection, mais avec différents degrés d'efficacité. Avec les requêtes dynamiques, l'utilisateur peut réutiliser un range-slider en ajustant la plage précédemment définie. Dans « Rolling the dice » [Elmqvist et al., 2008] l'utilisateur peut aussi ajuster une requête mais en utilisant des combinaisons de sélections ou en définissant une nouvelle sélection. Bien que cette interaction ne soit pas complètement explicitée, il semble que pour redéfinir une sélection, il faille la recréer complètement. Il n'est donc pas toujours possible de redéfinir la largeur d'une sélection rectangle, ou encore de modifier la forme créée par une sélection de type lasso. XmdvTools permet à l'utilisateur d'ajouter de nouveaux brushes mais ne permet pas de modifier des éléments d'une sélection précédente [Ward, 1994]. TimeSearcher permet de sélectionner des séries temporelles avec des rectangles de sélection ajustables en taille et position [Hochheiser & Shneiderman, 2004].

## **2. Opérations sur les sélections**

Un ensemble de rangesliders combine implicitement leurs requêtes en une seule en implémentant une opération booléenne ET : « Sélectionne les films entre les années 1980-1985 » (premier range slider) ET « sélectionne les films dont la durée est entre 1h30-1h40 » (deuxième range slider). Certains systèmes utilisent des sélections multiples à l'aide de « couches » souvent différenciées par des couleurs. L'utilisateur est alors capable de trouver des patterns entre les différents groupes colorés. La combinaison des sélections est alors réalisée avec un ensemble de formes colorées. Pour isoler des objets, l'utilisateur est alors obligé de réaliser une opération XOR : « je sélectionne un ensemble d'objets, je les copie en mémoire, j'applique l'opérateur XOR pour les faire disparaître de la vue courante ». L'utilisateur utilise aussi le ET logique pour combiner deux sélections en une seule. Young [Young & Shneiderman, 1993] a proposé une visualisation pour simplifier la compréhension des combinaisons des opérateurs logiques, mais cette dernière demande encore une charge cognitive importante pour être comprise.

Dans certains systèmes, l'utilisateur peut définir explicitement comment combiner sa sélection en choisissant les opérateurs booléens à appliquer : le résultat de l'opération crée alors une nouvelle couche avec une nouvelle couleur de sélection. L'interaction utilise alors des outils spécifiques [Ward, 1994], ou encore des interacteurs dédiés comme des boutons [Martin et al., 1995]. « Rolling the dice » [Elmqvist et al., 2008] réifie la sélection dans l'utilisation d'une pile de couches qui peuvent être combinées avec

un drag and drop. Le choix de l'opération booléenne est fait en fonction du bouton de la souris qui a permis de faire le « drag and drop » : le bouton droit applique un ET logique entre deux couches, alors que le bouton gauche applique un OU logique. Une fois l'opération exécutée, les deux couches fusionnent en une seule sans pouvoir les réutiliser individuellement.

## **D. Conclusion sur l'exploration de données**

---

Le modèle de Data Flow permet, d'une part, de structurer le processus de transformation des données vers une visualisation et, d'autre part, de détailler les interactions de l'utilisateur. Ainsi, le processus d'exploration de données s'appuie sur deux composants : la visualisation et l'interaction. Pour être efficace, la visualisation doit transmettre des informations assimilables par l'utilisation, et l'interaction doit être rapide et simple. La partie de cette thèse « réalisation de visualisation » illustre notre démarche pour la production de visualisations efficaces.



## I. Caractérisation de visualisations dans le domaine du Contrôle Aérien

Dans cette partie, nous allons utiliser les outils du domaine de l'InfoVis pour décrire des visualisations du domaine de l'ATC.

### A. Caractérisation des visualisations du domaine de l'ATC

Généralement, les modèles de visualisation décrivent des systèmes dédiés à l'exploration d'informations. Les visualisations ATC sont différentes des systèmes de visualisation classiques sur plusieurs points :

- Ces visualisations font partie d'une classe de visualisation appelée supervision. Une supervision affiche des informations à l'utilisateur sans que celui-ci ait la possibilité de les modifier. Il regarde les informations en les supervisant.
- Les données affichées sont dynamiques : elles évoluent au cours du temps, ce qui se traduit par des changements de propriétés des entités visuelles.
- Ces visualisations sont des systèmes critiques avec une forte pression temporelle.
- Les visualisations du domaine de l'ATC se ressemblent car elles se basent sur les mêmes données, ce qui oblige d'employer un modèle de description avec un niveau de précision suffisant pour différencier leurs caractéristiques.

Pour ce dernier point, les informations affichées proviennent soit des « données radar », soit du « plan de vol » de l'avion (succession de points de référence et de niveaux de vol que doit suivre l'avion pendant son vol). Nous considérons que les données sont stockées dans des tables dont chaque ligne contient un enregistrement. Le tableau suivant illustre un jeu de données simplifié pour les données radar.

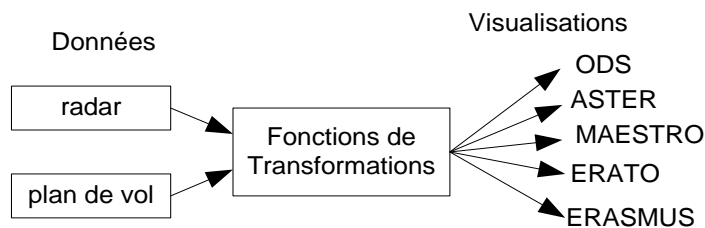
Données	Type
L'identifiant de l'avion	N
Position Longitude	Q ou Qlon
Position Latitude	Q ou Qlat
Vitesse Sol	Q
Niveau de vol (altitude)	Q
Vitesse verticale	Q
Heure du plot radar	Q

Le tableau suivant illustre un jeu de données simplifié pour les données plan de vol.

Données	Type
Identifiant de l'avion	N
Balise	N
altitude	Q
Heure de passage sur la balise	Q

A titre d'exemple, nous allons appliquer la caractérisation de C&M à plusieurs visualisations du domaine de l'ATC qui utilisent les mêmes données (radar et plan de vol):

- ODS (Operational input and Display System), visualisation radar,
- ASTER (Assistant for TERminal Sectors) [Benhacene, 2002], visualisation radar,
- MAESTRO (Moyen d'Aide pour Ecoulement Séquencé du Trafic Avec recherche d'Optimisation), séquenceur de trafic,
- ERATO (En Route Air Traffic Organizer), agenda de prévision de conflits entre avions.



Pour simplifier notre approche, nous considérons dans un premier temps que toutes ces visualisations sont statiques et nous nous intéressons uniquement à caractériser les entités visuelles principales de chaque visualisation (celles qui permettent à l'utilisateur de répondre à leur tâche principale). Ce choix de simplification est le même que le choix fait par C&M [Card & Mackinlay 1997]. Par exemple pour les vues radar, la tâche principale des contrôleurs est de séparer spatialement les avions, ils ont donc besoin de visualiser leurs positions. La fonction principale des visualisations radar est donc la fonction qui affiche à l'écran la position de l'avion, mais il n'est pas satisfaisant de considérer l'image radar avec cette seule fonction. Cette visualisation permet, entre autres, d'afficher les conflits, la vitesse, la tendance et l'altitude des avions. La caractérisation de la fonction de transformation principale n'est donc pas suffisante pour décrire entièrement une visualisation, c'est pourquoi nous avons caractérisé toutes les fonctions de transformation de l'image radar dans le chapitre « la fonction de transformation généralisée ». De plus certaines informations sont difficilement caractérisables car elles sont « émergentes ». C'est par exemple la direction de l'avion, transmise par la courbure de la comète radar. Cette notion d'émergence sera définie et étudiée dans les chapitres suivants.

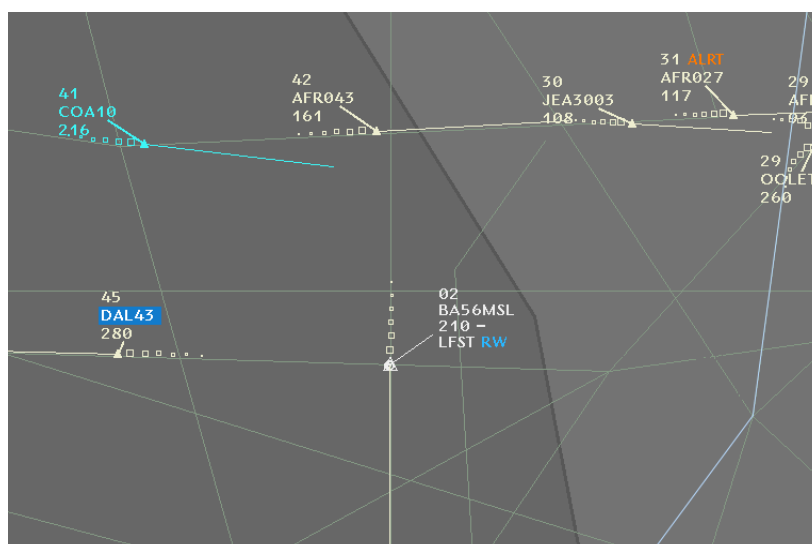
Le tableau suivant énumère les légendes que nous utilisons pour la caractérisation des visualisations avec le modèle de C&M.

L	Ligne (line)
S	Taille (size)
Sh	Forme (shape)
F	Fonction
N, O, Q	Nominal, Ordonné, Quantitatif
Lon, Lat	Longitude, Latitude
Pt	Point
Orien	Orientation
T	Texte
Lum	Luminosité
C	Couleur, teinte

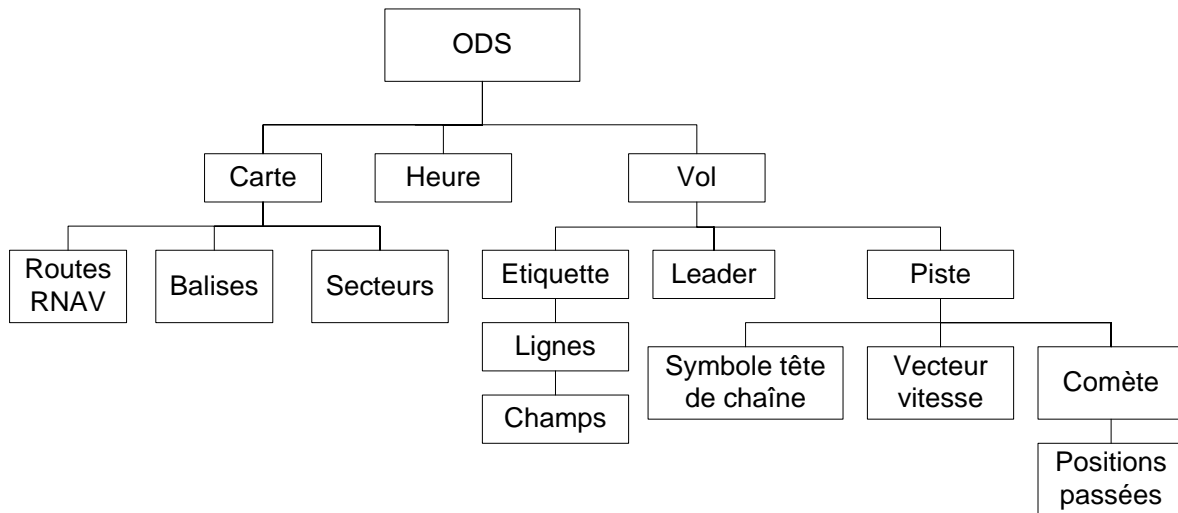
## B. Image radar : ODS

---

L'image radar ODS est la visualisation principale utilisée par les Contrôleurs Aériens. Elle représente la position des avions dans l'espace aérien en vue de dessus.



Une scène graphique ODS peut être considérée comme une hiérarchie d'entités regroupant des formes graphiques : l'étiquette comportant des informations sur le vol, les points RNAV (balises de radionavigation définissant les routes que doivent suivre les avions), les fonds de cartes avec les secteurs de contrôle etc... L'arbre suivant décrit la structure de la visualisation ODS. Cette structure illustre la notion d'assemblage utilisée par Wilkinson pour décrire les visualisations [Wilkinson, 1999].



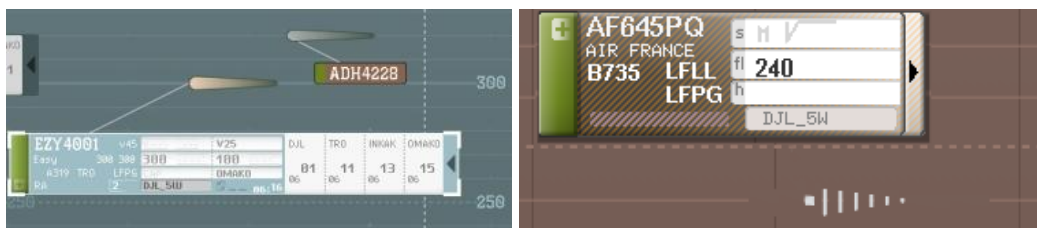
Chacune de ces entités visuelles est construite à l'aide de données brutes et d'une fonction de transformation. Nous allons caractériser la fonction principale d'ODS : l'affichage de la position des avions « vue du dessus ».

La position d'un avion est représentée par une forme (un triangle) positionnée selon la latitude et la longitude de l'avion. La fonction de transformation  $f$  est une simple mise à l'échelle et un centrage sur l'écran. La représentation de la position de l'avion a comme implantation le point au sens de Bertin avec une forme de triangle non orientée. Cette entité visuelle est appelée le symbole tête de chaîne et est caractérisée par le tableau suivant :

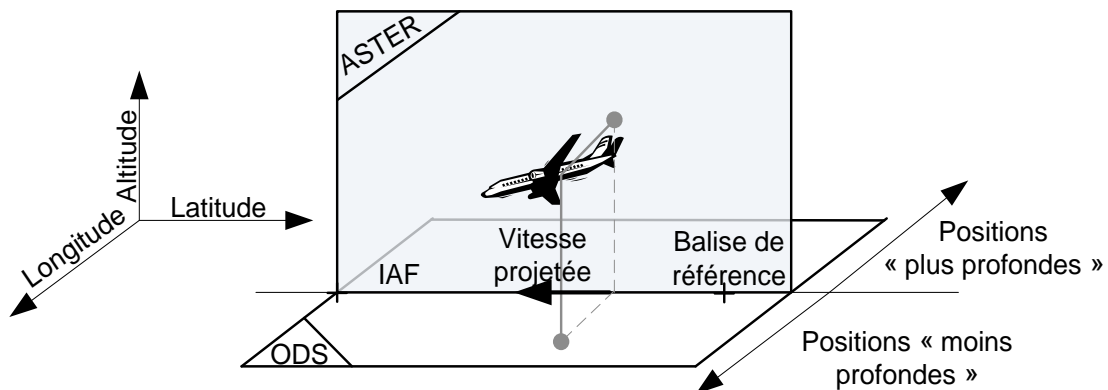
Data	D	F	D'	X	Y	Z	T	R	-	[]	CP
X	Q Lon	f	Q Lon	P				Forme			
Y	Q Lat	f	Q Lat		P						

### C. Image radar verticale : ASTER

L'outil ASTER [Benhacene, 2002] a été initialement conçu pour aider les Contrôleurs Aériens dans leur activité de régulation du trafic dans les secteurs terminaux. Ces secteurs gèrent à l'arrivée des avions avant leur prise en charge par les aéroports. En d'autres termes, ASTER permet aux contrôleurs de prévoir et de séquencer les atterrissages des avions en anticipation. De plus, il renseigne les systèmes du séquençement des clairances (ordres) données aux avions.

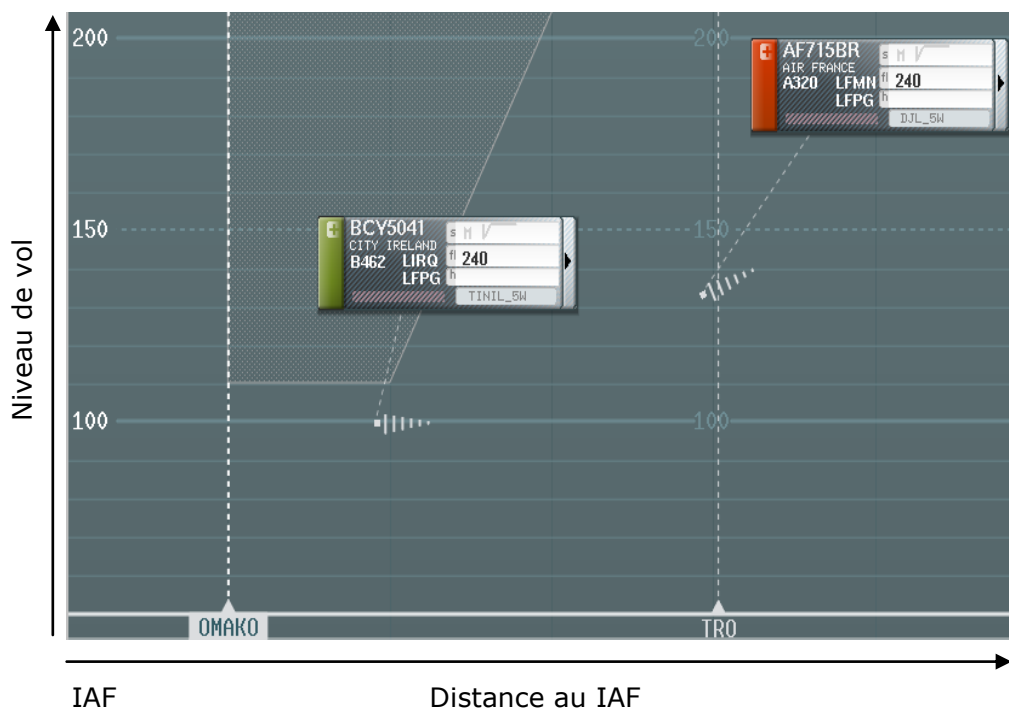


Dans ces secteurs, les avions se dirigent tous vers un point de sortie du secteur appelé IAF (Initial Approach Fix). Les avions ainsi contrôlés sont quasiment tous en descente, c'est pourquoi ASTER utilise une vue verticale du trafic.



ASTER affiche le niveau de vol (AFL, Actual Flight Level) sur l'axe des ordonnées des avions et la distance au IAF (point de radionavigation pour la sortie de secteur) sur l'axe des abscisses.

Pour afficher la position d'un avion avec ASTER, le système calcule sa position par rapport à sa projection sur un axe de référence. Cet axe passe par le point de sortie du secteur (IAF) et par une balise de référence (appelée CHABI) de telle sorte que cet axe coupe en deux le secteur dans sa partie la plus longue.



Une forme graphique (dite « comète ») représente la position de l'avion issue des données radar. La comète est placée à l'écran grâce à une fonction de projection sur l'axe IAF, balise de référence pour l'axe des X et l'altitude l'avion pour l'axe des Y. La profondeur de la comète est donnée par son occlusion par rapport aux autres comètes et une luminosité (sombre pour les comètes très distantes de la ligne de référence, clair



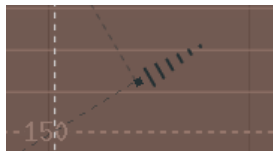
pour les comètes les plus proches). Le Z du tableau de caractérisation est donc la profondeur du plan ; elle correspond à l'occlusion entre les entités visuelles et fournit ainsi une information ordonnée (O) partielle (des objets les plus profonds, aux objets les plus proches). Cette information est partielle, car la perception de l'ordonnement des comètes par la profondeur n'est effective que si les comètes se chevauchent. La taille des comètes indique la vitesse de l'avion et son orientation indique sa direction dans le plan vertical.

Data	D	F	D'	X	Y	Z Occlusion	T	R	-	[]	CP
Plot	Lat Lon (OxQ)	f	distance	P				Forme			
altitude	altitude	f	Q		P						
Taux de descente	Vitesse	f	Q					Orient ation			
Profondeur	Profondeur	f	O(partiel)			P		Lum			

Le taux de descente de l'avion est exprimé par les Contrôleurs Aériens en pieds par minute (ft/min). La distance de la tête à la queue de la comète d'ASTER correspond à la distance projetée parcourue par l'avion en une minute de vol. L'axe vertical étant gradué en pieds, le contrôleur aérien peut directement lire la vitesse verticale de l'avion en comparant les valeurs des niveaux de vol que coupe la comète :



Le taux de descente de cet avion est de 2000 ft/min



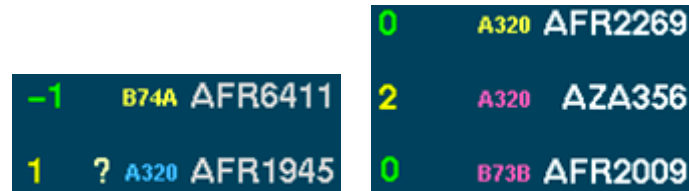
Le taux de descente de cet avion est de 1000 ft/min

## D. Séquenceur des avions à l'atterrissage : MAESTRO

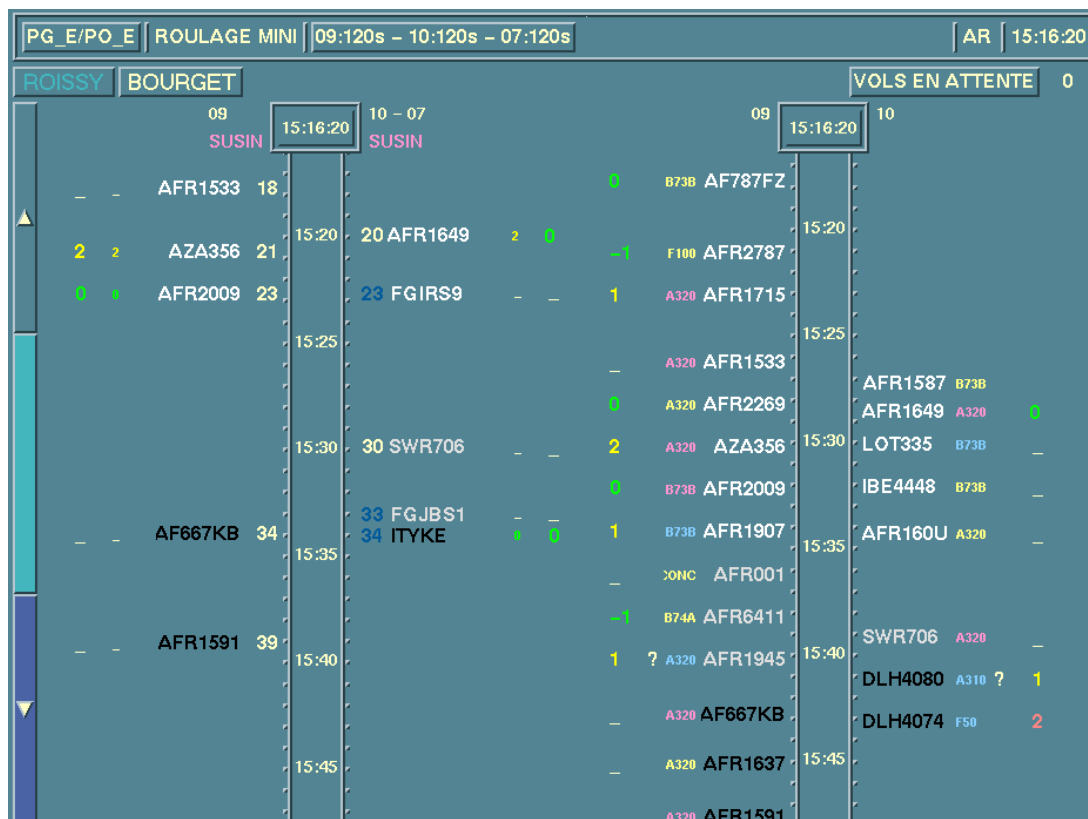
MAESTRO est un logiciel dédié à la régulation du Trafic Aérien au niveau des aéroports. Il permet de créer des séquences d'avions pour l'atterrissage. Il est ainsi possible de prévoir les altérations de vitesse (accélérations, décélérations) à appliquer aux avions pour respecter les cadencements prévus par les Contrôleurs Aériens. La fonction principale de MAESTRO est de positionner des avions dans une séquence ordonnée chronologiquement pour l'atterrissage sur une piste. L'image suivante montre deux colonnes qui correspondent aux pistes de Roissy et du Bourget avec des listes d'avions ordonnées dans le temps.

Cette visualisation donne des informations quantitatives (Q). Les avions sont positionnés à une distance propositionnelle au temps restant avant l'atterrissage (c'est un temps estimé). De plus l'information est codée textuellement avec l'affichage de l'heure et de gradations pour permettre à l'utilisateur de lire l'heure prévue d'atterrissage du vol.

Des chiffres sont inscrits à coté des indicatifs des avions. Ces chiffres correspondent au nombre de minutes à perdre ou à gagner pour chaque avion afin respecter la séquence (-1 signifie à faire perdre une minute de vol, c'est-à-dire accélérer). La couleur des chiffres permet de coder si le nombre de minutes est à perdre ou à gagner (vert : positif, jaune : négatif). Les valeurs négatives sont deux fois plus longues en taille à l'écran que les valeurs positives car elles utilisent le signe « - ». Il en résulte un effet de mise en avant des valeurs négatives car elles sont décalées par rapport aux valeurs positives : la ligne « Altération temporelle » est codée par la couleur et par la variable visuelle X écran.



L'image suivante montre l'interface complète de MAESTRO.

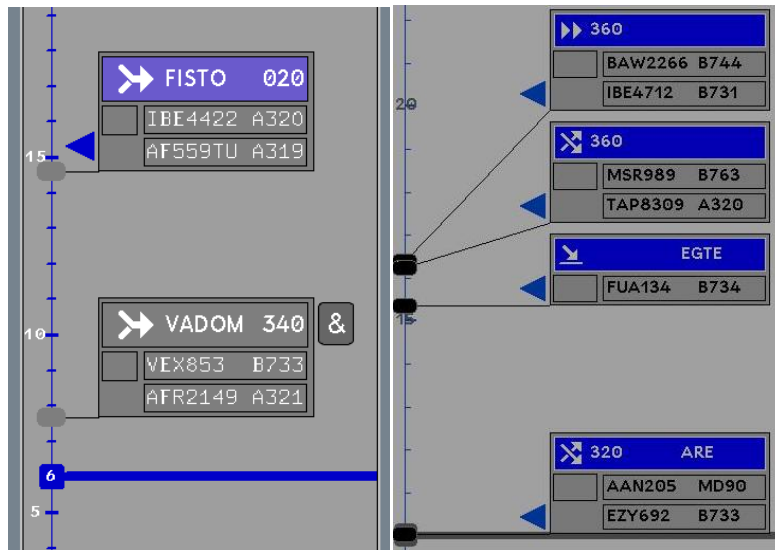


Le tableau suivant décrit la caractérisation de MAESTRO.

Data	D	F	D'	X	Y	Z	T	R	-	[]	CP
Piste d'atterrissage	N		N	N colonne							Texte
Heure d'atterrissage	Q		Q		P						
Altération temporelle	Q		Q	P				C			
Nom	N		N		P						

## E. L'agenda des conflits : ERATO

La tâche principale des Contrôleurs Aériens est d'assurer une distance minimale de sécurité entre les avions. Le passage potentiel de deux avions en dessous des normes de séparation verticale et horizontale est nommé un conflit. Ainsi, la fonction principale de l'agenda ERATO est d'afficher une liste chronologiquement quantitative des conflits futurs détectés par le contrôleur ou la machine. L'heure prévue du conflit (t) est notée par un point relié à étiquette détaillant les données importantes.



La liste des conflits est affichée sur une échelle de temps ; elle code donc une information quantitative Q (ce conflit va se produire dans deux fois moins longtemps que celui-ci). Les étiquettes sont elles aussi reliées par une ligne à cette échelle. Dans la majorité des cas, la position des étiquettes donne elle aussi une information quantitative, mais lorsqu'elles sont trop nombreuses, le système d'anti-recouvrement les décale. Ainsi elles perdent leur caractère quantitatif Q pour ne coder qu'une information ordonnée O (ce conflit est avant celui-ci et après celui-là).

Data	D	F	D'	X	Y	Z	T	R	-	[]	CP
Temps	Q	f	Q		P						
Étiquette	Q		Q ou O						L		T
Type de conflit								sh			

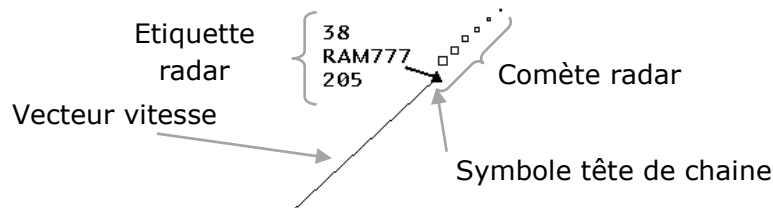
## F. Discussion sur la caractérisation

La caractérisation de la fonction de transformation principale n'est pas suffisante pour caractériser une visualisation car son choix est subjectif. Il est dépendant du point de vue de l'utilisateur pour le choix de l'information principale affichée par la visualisation. On peut donc à juste titre vouloir caractériser toutes les informations contenues dans un visuel.

A titre d'exemple, nous proposons de caractériser tous les éléments visuels important de la piste radar.

## 1. Caractérisation de la piste radar

La piste radar affiche les positions passées de l'avion avec la comète. La position courante de l'avion est affichée par le symbole tête de chaîne. Les informations concernant l'avion sont stockées dans l'étiquette radar. La vitesse de l'avion est donnée par le vecteur vitesse.



### a) La comète ODS

La comète ODS est constituée de plusieurs carrés qui correspondent aux positions passées de l'avion. La juxtaposition des carrés forme un nouvel élément visuel qui apporte de nouvelles propriétés perceptibles (les carrés se groupent par l'effet de proximité de la Gestalt [Kofka, 1935]). Le modèle de C&M ne peut pas directement s'appliquer aux comètes car cette sur-forme génère de l'information qui n'est pas disponible dans les données brutes pour dessiner la comète : il n'est pas utile de disposer de l'information de tendance horizontale (l'avion a tendance à tourner à droite ou à gauche) pour afficher une comète qui est courbée, cette information émerge du design. L'image suivante montre un avion qui tourne alors que pour dessiner cette comète le système n'utilise que des informations de position et pas une donnée « virage » :



Ce design de comète ainsi que d'autres choix de représentation des positions des avions seront étudiés en détail dans le chapitre sur la caractérisation des comètes.

### b) L'étiquette radar

L'étiquette est un groupe formé par l'alignement vertical des lignes. Pour la caractériser, nous avons besoin de définir précisément la propriété d'encapsulation :

- Deux entités graphiques sont groupées quand il existe une relation contenant-contenu ([ ] c-c) entre elles. Par exemple les lignes de l'étiquette radar sont des filles de l'étiquette. Il s'agit de l'encapsulation [ ] définie par C&M.
- L'étiquette est constituée de lignes de texte qui sont groupées par leur justification à gauche. La couleur et la taille peuvent aussi être des variables permettant cette agrégation et font apparaître une relation de groupe ([ ] g). Ce

phénomène s'opère lorsqu'au moins deux entités visuelles ont une variable visuelle de même valeur : X ou Y, couleur...

Data	D	F	D'	X	Y	Z	T	R	-	[] g	cp
Vitesse	Q		Q	O Champs	O ligne					Alignement	Texte
Nom	N		N	O	O+1 ligne						
altitude	Q		Q	O	O+2 lignes						
Tendance	Q		Q	O+1	O+2 lignes						
Balise de sortie	N		N	O	O						

### c) Le symbole tête de chaîne

Le symbole tête de chaîne (la position de l'avion) est l'information principale utilisée par les Contrôleurs Aériens. Nous l'avons déjà caractérisé.

Data	D	F	D'	X	Y	Z	T	R	-	[]	CP
X	Q Lon	f	Q Lon	P				Forme			
Y	Q Lat	f	Q Lat		P						

### d) Le vecteur vitesse

Le vecteur vitesse est une ligne qui commence avec le symbole tête de chaîne et finit à la position future de l'avion si celui-ci garde la même vitesse et la même direction. Nous utilisons la notation de Bertin pour qui la ligne a une taille (épaisseur) et une longueur, et dans le cas du vecteur vitesse, cette longueur est proportionnelle à la vitesse de l'avion. L'orientation de cette ligne est la même que la direction de l'avion. La caractérisation du vecteur vitesse est donnée dans le tableau suivant :

Name	D	F	D'	X	Y	Z	T	R	-	[]	CP
Longitude	Q	F	Q	L				Ligne			
Latitude	Q	F	Q		L						
Vitesse	Q	F	Q					S			
Direction	Q	F	Q					O			

### e) Le leader

Le leader est une ligne qui part du symbole tête de chaîne et arrive près de l'étiquette radar.

Data	D	F	D'	X	Y	Z	T	R	-	[]	CP
Position Tête de chaîne	Q,Q	f	Q,Q	P	P						
Position Etiquette	Q,Q	f	Q,Q	P	P						
Couple Etiquette Tête de chaîne	N	f	N						L		

Nous avons ainsi caractérisé avec plusieurs tableaux un seul élément de la visualisation radar. La section suivante propose de synthétiser ces informations dans un seul tableau.

## G. La fonction de transformation généralisée

La caractérisation de la fonction principale avec le modèle de C&M n'est pas suffisante. Dans cette section, nous cherchons à caractériser les visualisations de la façon la plus précise et la plus exhaustive possible. Nous proposons donc d'utiliser la notion de fonction de transformation généralisée qui va expliciter comment des données brutes sont transformées par une fonction mathématique pour réaliser un élément visuel à l'écran. Cette caractérisation est une version similaire des travaux de Baudel [Baudel, 2002] et Wilkinson [Wilkinson, 1999].

La méthode consiste à recenser chaque forme graphique importante pour l'activité. Pour chaque entité graphique, il faut déterminer les informations qu'elle transmet. Le tableau suivant caractérise la comète ASTER avec la fonction de transformation généralisée

	Codage	Fonction
<b>Comète ASTER</b>	Forme	Point (X,Y,taille,orientation,forme)
Vitesse	Taille	
Vitesse verticale	Orientation	

Le tableau suivant est une synthèse de la caractérisation des entités graphique d'ODS avec le vecteur vitesse, le symbole tête de chaîne, la comète et l'étiquette. La première colonne comporte toutes les données classées par entité graphique, la deuxième colonne donne l'implantation et les variables visuelles utilisées. Enfin, la dernière colonne décrit une fonction pour synthétiser chaque entité graphique : Ligne, Point, Zone. Par exemple, pour le symbole tête de chaîne cette fonction est : Point (X, Y, forme, couleur) avec X et Y la position en latitude et en longitude de l'avion et la forme un triangle.

	Codage	Fonction
<b>Vecteur Vitesse</b>	Ligne	line (X, Y, $X + \alpha.V_x, Y + \alpha.V_y$ $\alpha [1, 3, 6, 9]$ )
Vitesse de l'avion	Taille	
Cap de l'avion	Orientation	
Position de l'avion	Position	
Etat de l'avion	Couleur	
<b>Symbole tête de chaîne</b>	Point	Point (X, Y, forme)
Position de l'avion	Position	
Etat de l'avion	Couleurs et formes	
<b>Comète</b>	Liste de points -> Ligne	Point (X(t), Y(t), forme (size(t)), Pt(X(t-1), Y(t-1), forme (size(t-1)), ... Point(X(t-n), Y(t-n), forme (size(t-n)))
Vitesse de l'avion	Taille	
Positions passées	Positions	
Tendance de l'avion	Distance entre points	
Evolution du cap	Forme	
Etat de l'avion	Couleur	
<b>Etiquette</b>	Zone	Point (ligne, colonne, texte)
Vitesse	Texte	
Nom de l'avion	Texte	
altitude	Texte	
Tendance verticale	Texte	
Balise de sortie	Texte	

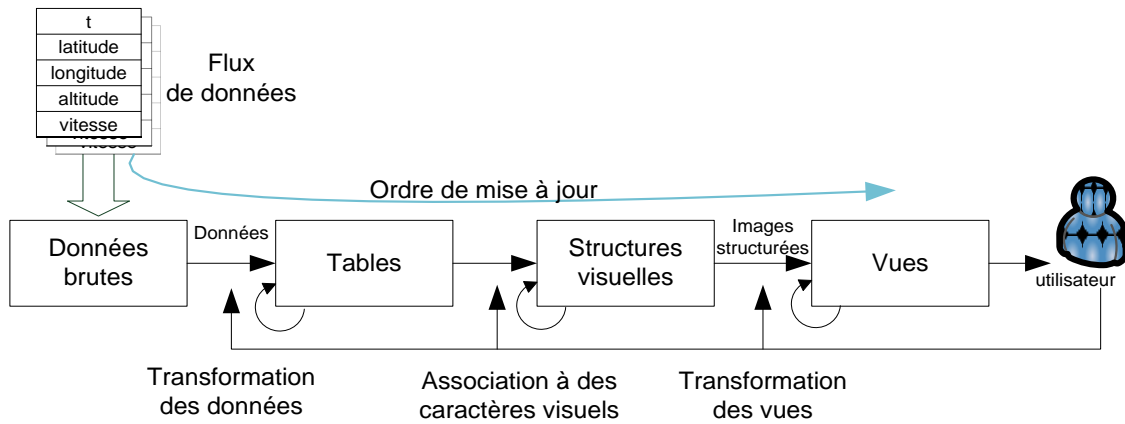
La réalisation de ce tableau permet de synthétiser toutes les informations qui sont affichées dans une visualisation. Cette approche est plus exhaustive que la caractérisation de C&M mais trouve ses limites dans son utilisation pour comparer les visualisations. Du fait de la taille de ce tableau nécessaire à la description de toutes les entités visuelles, l'utilisateur ne peut pas facilement comparer deux caractérisations comme il aurait pu le faire avec les caractérisations de C&M (caractérisation de la fonction principale de la visualisation). Il y a donc un compromis à trouver entre exhaustivité et compacité.

## H. Le temps et la dynamique de l'image

Pour simplifier notre démarche, nous avons décrit des visualisations statiques. Or les visualisations du domaine de l'ATC utilisent la dynamique de l'image pour coder de l'information. Par exemple, l'animation est utilisée lors de la détection d'un conflit par le système radar (passage potentiel sous la norme de séparation verticale et horizontale d'au moins deux avions). La caractérisation de l'animation est un élément important pour comprendre les informations que porte une visualisation [Schlienger et al., 2007] [Bartram et al., 2001].

### 1. Les sources de l'animation

Avec le modèle de Data Flow, l'animation peut être modélisée de plusieurs manières. C'est premièrement l'arrivée d'une nouvelle donnée (mise à jour des données) qui déclenche l'animation. En effet, il faut alors mettre à jour l'image et c'est cette succession d'images qui permet de percevoir les variations de données dans le temps.



Dans le cas d'ODS, cette mise à jour s'opère toutes les 4 secondes, ce qui a un impact sur la perception du contrôleur. Si au bout d'un intervalle de plus de 4 secondes la piste radar n'a pas été mise à jour, le contrôleur perçoit l'absence de changement et en déduit une perte de données radar. Le contrôleur s'attend alors à ce que le système lui indique cette information avec un symbole particulier (dans le cas d'ODS, un icône d'avion est affichée à la place du symbole tête de chaîne).

L'état d'une entité visuelle dans le temps est une autre source d'animation. Dans le cas d'une détection de conflit, le système fait clignoter l'étiquette des avions concernés. Il n'y a pas de nouvelle donnée, mais l'état des entités visuelles change avec par exemple la couleur de fond du texte de l'indicatif de l'avion qui change toutes les secondes. L'animation est une nouvelle variable visuelle [Athènes et al. 2000].



Le temps agit donc au niveau des données, et au niveau de l'image. Nous considérons que le temps est manipulable à ces deux niveaux. Par exemple, il peut être suspendu au niveau de la mise à jour des données. Le trafic est figé, mais pas la visualisation : les animations de l'image continuent de « tourner » (exemple : clignotement de l'étiquette). Le temps peut être choisi, par exemple avec un slider, afin de visualiser une situation de trafic passé. Cette visualisation donne toujours des informations par l'animation : la fréquence des clignotements éventuels n'est pas impactée par la manipulation du temps des données. L'image peut aussi être figée, sans arrêter la mise à jour des données. Les animations comme le clignotement des étiquettes sont alors arrêtées, mais les comètes continuent à se mouvoir.



Dans la taxonomie de C&M, la seule notion de temps décrite est celle de la représentation sur l'écran à un instant donné (la colonne T du modèle de C&M) : la visualisation n'est plus considérée comme une image uniquement statique mais comme un film (i.e. une séquence d'images). Le T de C&M peut être considéré comme le numéro de l'image (frame) dans le film dont l'utilisateur peut lancer le visionnage. Une façon de modéliser la dynamique du symbole tête de chaîne dûe aux mises à jour des données, est d'associer chaque donnée radar à un numéro d'image du film : visualiser le trafic consiste à jouer le film. Le tableau suivant explicite l'utilisation de la mise à jour des données pour créer l'animation. Le temps est une donnée brute quantitative qui est ensuite transformée en donnée ordonnée et utilisée pour afficher une information à un temps donné.

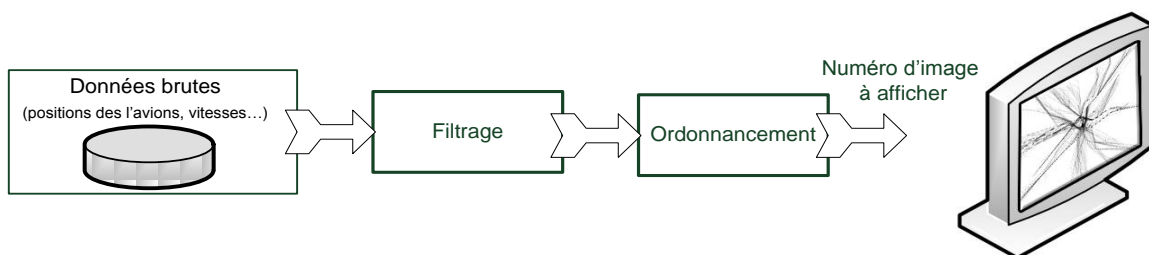
Data	D	F	D'	X	Y	Z	T	R	-	[]	CP
X	Q Lon	f	Q Lon	P				form e			
Y	Q Lat	f	Q Lat		P						
T	Q	f	O				P				

Nous renommons la colonne T du modèle de C&M en Fr pour indiquer le numéro de *frame* affiché à l'écran. Ce nouveau modèle fait aussi apparaître l'animation due à la mise à jour (Amaj). Enfin, nous ajoutons la colonne A pour les animations type Variables Visuelles indépendantes de la mise à jour des données (Animation sur X, Y, la couleur, la taille...).

Name	D	F	D'	X	Y	Z	Fr	Amaj	A	R	-	[]	CP

## I. Utilisation du modèle de la dynamique de l'image

Le schéma suivant explicite notre conception de l'animation avec le modèle de Data Flow et la dynamique de l'image. Les données brutes sont toutes les informations dont on dispose avant leur affichage. Cela implique, dans le cas des systèmes d'affichage en temps réel, que ces données soient mises à jour régulièrement et que, comme expliqué précédemment, cette mise à jour génère une animation. Ces données brutes sont filtrées pour ne choisir que celles qui seront affichées à l'écran. Par exemple dans le cas de l'écran radar, on veut afficher la position courante des avions ; ainsi la phase de filtrage va consister à sélectionner les données qui correspondent au temps courant. Ensuite, ces données sélectionnées sont ordonnées en fonction d'un critère. Dans le cas le plus commun, c'est le temps qui est utilisé pour fournir une séquence ordonnée d'images à afficher à l'écran.



Ce modèle nous permet d'identifier plusieurs cas d'interaction avec la dynamique de l'image :

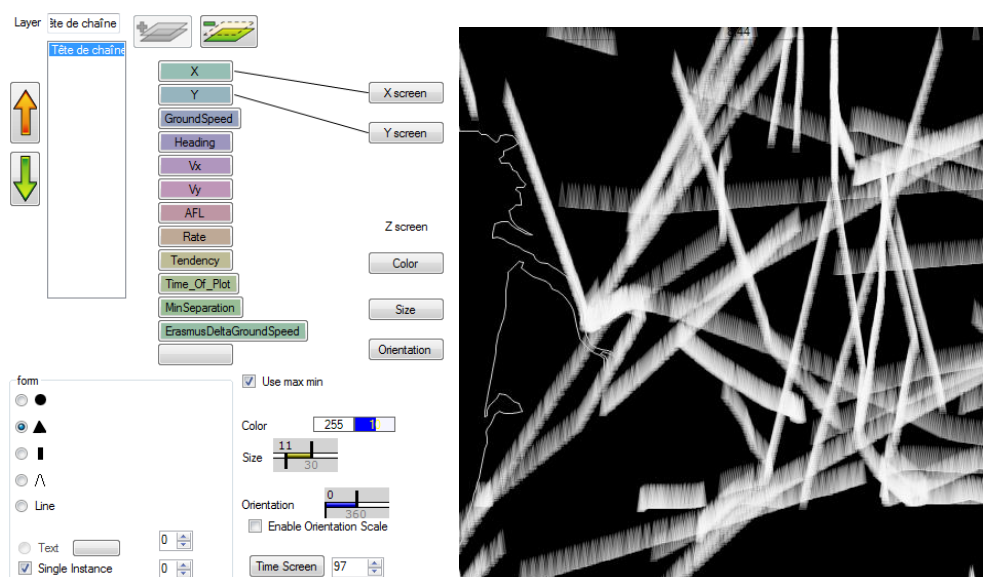
- Des actions sur le filtrage permettent de générer de l'animation avec par exemple le cas du symbole tête de chaîne qui correspond à des données filtrées sur le temps courant.
- Des actions sur l'ordonnement qui permet d'afficher des images qui ont une correspondance temporelle. Mais on peut imaginer d'autres ordonnancements avec par exemple l'altitude des avions. On obtiendrait alors une animation par couche des avions avec d'abord les avions qui sont au sol, pour finir avec les avions en vol à très haute altitude.
- Des actions sur le numéro d'image à afficher. Dans ce cas, c'est l'utilisateur qui modifie la séquence d'affichage et ainsi contrôle son déroulement par exemple dans le temps (cas d'un ordonnancement avec le temps), ou encore du niveau de vol à afficher (cas d'un ordonnancement par l'altitude des avions).

## 1. Implémentation

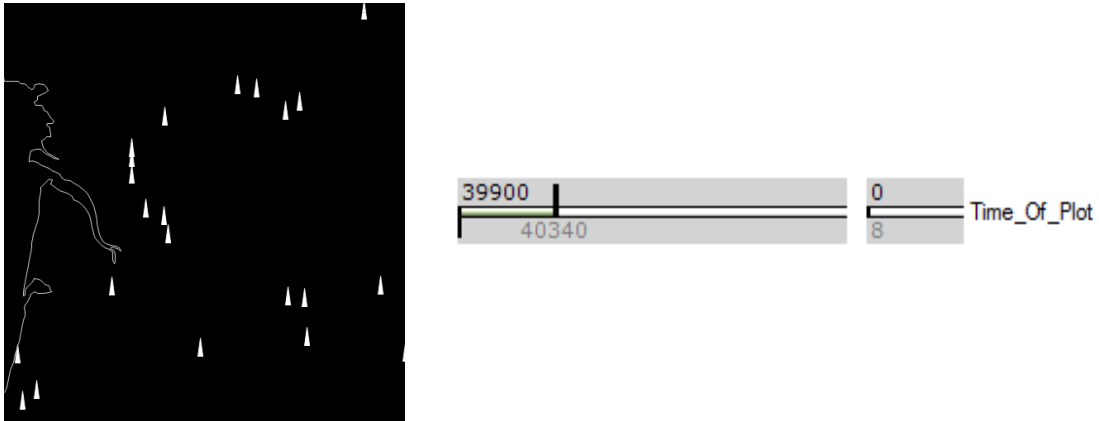
Afin de valider notre modélisation de l'animation, nous avons recréé l'ensemble des formes d'animation en utilisant les principes de numéro d'image, d'ordonnement et de filtrage. Pour ce faire, nous avons implémenté le modèle de Data Flow dans un logiciel. Cette démarche a permis :

- de décrire chacune des formes d'animation,
- d'utiliser le modèle de Data Flow pour créer l'animation,
- de valider notre modèle.

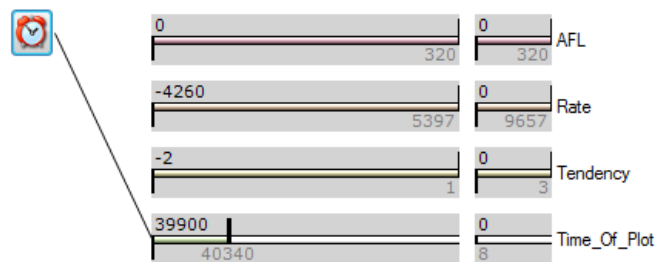
Nous créons donc une couche appelée « Tête de chaîne » qui fait correspondre la position de l'avion avec un triangle positionné à l'écran. L'image suivante montre les connexions entre les données brutes et les variables visuelles. L'image résultante est l'ensemble des données brutes traitées par la fonction de transformation ainsi décrite : l'affichage d'un triangle positionné en fonction des champs X et Y.



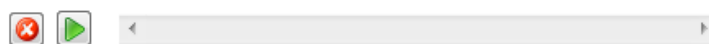
Ensuite, nous modifions les paramètres de filtrage pour n'afficher que les positions des avions à un instant donné. Le premier Rangeslider permet de choisir grossièrement la plage de filtrage, le deuxième permet d'être plus précis dans la première plage sélectionnée.



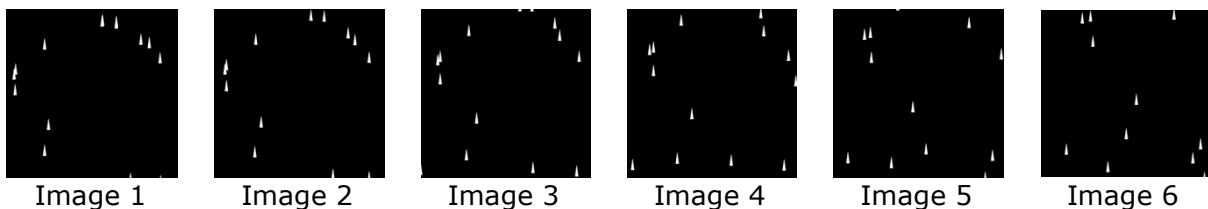
La dernière phase consiste à faire correspondre l'heure d'affichage à l'écran avec le numéro de frame des images ordonnées par le champ « time » des données.



L'utilisateur peut alors soit jouer l'animation avec le bouton Play, soit contrôler cette animation avec un Slider (contrôle du numéro d'image à afficher).



L'animation produit bien le déplacement des positions des avions dans le temps.



L'utilisateur peut faire d'autres choix pour l'animation comme par exemple la connexion du numéro de l'image avec le niveau de vol. L'animation affiche d'abord les avions qui sont au sol et finit par ceux qui ont la plus haute altitude. D'autres choix d'animation sont possibles comme l'orientation des avions, leur vitesse verticale, etc...

## 2. Animation de type "interpolation"

Nous avons précédemment défini l'animation comme une succession d'images qui correspondent à des sous-ensembles des données: pour créer l'animation des positions des avions, les données sont filtrées avec le temps. L'animation résultante n'est pas toujours fluide : des saccades sont perçues lorsqu'il existe une discontinuité dans la période d'enregistrement des données. L'animation avec interpolation des données permet de supprimer la perception de ces saccades.

Pour réaliser l'animation avec interpolation, le système doit être renseigné avec un identifiant unique pour chaque entité visuelle pour pouvoir faire l'interpolation des variables visuelles entre deux images. Pour illustrer l'animation interpolée, nous avons répliqué des travaux précédents [Schlienger et al., 2007] avec un jeu de données contenant une liste d'indicatifs d'avions avec deux ordonnancements différents.

pos	1,00	2,00	3,00	4,00	5,00
time	0,00	0,00	0,00	0,00	0,00
Name	AF 1273	DHL 512	BEE 7031	BAW 314	RAM 762

pos	1,00	4,00	2,00	3,00	5,00
time	1,00	1,00	1,00	1,00	1,00
Name	AF 1273	DHL 512	BEE 7031	BAW 314	RAM 762

Il faut indiquer au système comment afficher cette liste : le texte est positionné en fonction du champ pos.

<b>Location</b>	
LocationX	
LocationY	<b>pos</b>
LocationZ	

Il faut ensuite renseigner le système sur le champ qui permet d'identifier chaque élément visuel pour lui permettre de faire l'interpolation entre deux sous-ensembles de la base de données.

<b>Animation</b>	
AnimationStepsNr	<b>10</b>
KeyField	<b>Name</b>

Finalement, il nous reste à indiquer que nous souhaitons animer cette liste avec le filtrage sur le champ time.

<b>Frame</b>	
Frame	<b>time</b>
Shift	<b>0</b>

L'animation peut se produire avec le déplacement de trois éléments dans la liste.

RAM 762	RAM 762	RAM 762	RAM 762	RAM 762	RAM 762
BAW 314	BAW 314	BAW 314	<del>BAW 314</del>	DHL 512	DHL 512
BEE 7031	<del>BEE 7031</del>	<del>BEE 7031</del>	BEE 7031	<del>BAW 314</del>	BAW 314
DHL 512	<del>DHL 512</del>	<del>DHL 512</del>	BEE 7031	BEE 7031	BEE 7031
AF 1273	AF 1273	AF 1273	AF 1273	AF 1273	AF 1273

Avec notre implémentation de l'animation par interpolation, nous avons validé un certain nombre de points [Heer & Roberstson, 2007] :

- L'interpolation des variables visuelles fonctionne avec notre modèle.
- L'interpolation peut s'effectuer sur toutes les variables visuelles quantitatives et ordonnées (couleur, taille), mais pas sur la forme (morphing), ou la texture.
- Des éléments additionnels sont requis pour simuler toutes les classes d'animation comme le séquençement des animations, le choix de la vitesse d'animation...

## **J. Validation et utilisation du modèle de DataFlow**

---

La validation et l'utilisation du modèle de DataFlow ne contient pas de contribution au domaine de la recherche. Par contre, cette démarche nous a permis de valider l'utilisation de ce modèle et il servi de support à notre recherche. Nos travaux, avec l'implémentation du modèle de DataFlow, sont disponibles dans l'annexe de cette thèse. Notre démarche à été la suivante :

- implémentation de la transformation des données en entités visuelles,
- implémentation de l'interaction avec les données (filtrage),
- réalisation de visualisations (la marche de napoléon et la vue radar).

Les conclusions de notre démarche sont les suivantes :

- nous avons rendu explicite les choix de design pour la réalisation de visualisations,
- les choix de design forment une caractérisation de la visualisation : c'est le contenu sans la mise en forme,
- pour afficher les visualisations du domaine ATC, le modèle de DataFlow doit être amélioré avec l'ajout de cases mémoires, du filtrage dynamique, de l'ordonnancement (pour gérer l'occlusion), de gestion de l'animation,
- nous avons mis en évidence les problématiques liées à l'utilisation de données contenant des erreurs et des incertitudes (taux d'échantillonnage variable, champs vides),
- nous avons réalisé des nouveaux designs.

## **K. Conclusion**

---

Dans ce chapitre, nous avons montré des exemples de caractérisation de visualisations. Notre but initial était de montrer la faisabilité de la caractérisation et d'identifier les difficultés de cette démarche. Nous avons donc commencé par l'application directe du modèle de C&M pour décrire de façon compacte des visualisations issues du domaine de l'ATC. Ensuite, nous avons étendu cette caractérisation en ajoutant la dynamique de l'image.

## **1. Conclusion sur l'application du modèle de caractérisation aux visualisations issues de l'ATC**

Nous avons essayé d'appliquer le modèle de C&M à différents types de visualisation du domaine de l'ATC [HURTER 07] [HURTER DSVIS 08] [HURTER ICRAT 08]. Nous avons évalué avec des critères quantitatifs son efficacité à afficher les informations transmissibles par les visualisations. La caractérisation de la fonction de transformation principale des visualisations permet de générer des tableaux qui synthétisent les informations importantes contenues dans la visualisation. Ces résultats montrent que le modèle de C&M est adapté à la caractérisation des fonctions principales de ces visualisations. Mais il est à nuancer sur plusieurs points :

- Caractériser une visualisation ne peut pas se limiter à la caractérisation de sa fonction de transformation principale, car beaucoup d'autres informations visuelles sont disponibles.
- Certaines entités visuelles sont simples, comme par exemple l'affichage de l'heure, et ne présentent pas d'intérêt à être caractérisées.
- La caractérisation de C&M ne prend pas en compte l'aspect dynamique des visualisations.

Bien que le modèle de C&M rende possible la caractérisation des visualisations et leur comparaisons (avec la comparaison de leur tableau de caractérisation), ce modèle ne permet pas d'être exhaustif. En effet, nous avons décrit la structure visuelle de la visualisation d'ODS avec une hiérarchie entre les entités visuelles. Nous aurions dû établir une caractérisation pour chacune de ces entités visuelles. Ceci est fastidieux et nuirait sans doute à un des buts recherchés de la caractérisation : la compacité et la facilité de lecture. De plus, le Modèle de C&M ne permet pas de rendre compte des visuels dits « émergents » : par exemple, le modèle de C&M ne peut pas caractériser toutes les informations qui sont codées dans la comète radar, comme la direction de l'avion.

## **2. Conclusion sur la dynamique de l'image**

Comme nous l'avons déjà évoqué en introduction, la dynamique de l'image est très importante et est très largement utilisée dans les visualisations. Elle permet de coder de nombreuses informations que nous n'avons pas pu recueillir dans le modèle de caractérisation. A titre d'exemple, la vitesse d'apparition des conflits dans ERATO permet de prévenir le contrôleur que la complexité de gestion du trafic va fortement augmenter.

Nous avons identifié les différentes sources de la dynamique de l'image :

- la mise à jour des données,
- le numéro d'image affichée,
- les animations de type Variable Visuelle.

Nous avons validé ces trois sources de la dynamique de l'image avec l'implémentation d'un logiciel.

Enfin, nous avons rajouté au modèle de C&M les trois colonnes correspondantes pour permettre la caractérisation de la dynamique de l'image. Nous avons renommé la colonne T du modèle de C&M en Fr car il s'agit bien du temps du film. Nous avons ajouté une colonne A pour l'animation des données indépendantes du modèle de Data Flow. Finalement, nous avons identifié le processus de mise à jour automatique de la visualisation avec l'arrivée de nouvelles données (colonne Amaj).

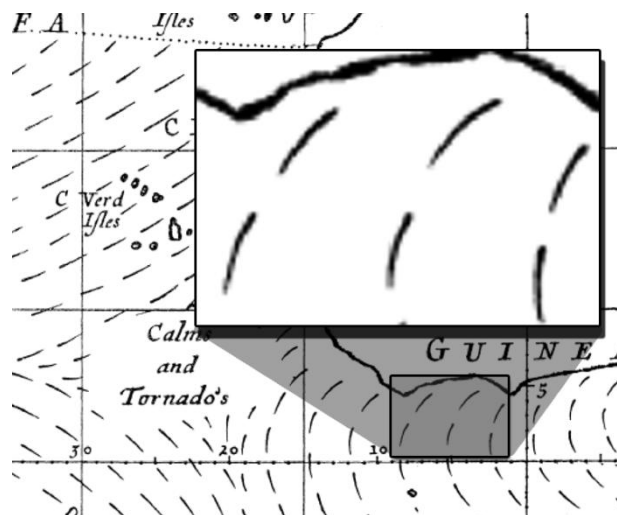
## II. Comparaison des comètes radar

Dans cette partie, nous proposons d'appliquer nos connaissances sur l'analyse et la compréhension des représentations graphiques au domaine spécifique du Contrôle Aérien en se focalisant sur l'outil principal du contrôleur aérien : l'écran radar. Dans cette classe de visualisations, la position de l'avion est toujours affichée par une comète mais avec des choix de design différents. Nous proposons donc d'analyser les designs utilisés pour afficher la position de l'avion.

Notre but est de synthétiser les designs en les caractérisant. A partir de cette caractérisation, nous allons être capables de comparer ces designs, de les opposer. Cette analyse des designs va ainsi permettre de mettre en avant les avantages et les inconvénients des choix de représentation. Cette analyse va mettre en œuvre les connaissances dans le domaine de la perception. Elle permettra ainsi de juger l'efficacité de la transmission d'informations en fonction des choix de représentations.

### A. Les comètes radars dans le domaine de l'ATC

L'outil principal pour les contrôleurs aériens est l'écran radar qui affiche en temps réel les positions des avions avec une comète radar. Avant de détailler les propriétés visuelles de cette comète, il est important de comprendre les origines de ce design. Les propriétés visuelles de la comète ont été utilisées pour la première fois au 17e siècle par Edmond Halley [Norman & Thrower, 1969] pour afficher la direction des vents de la mousson avec une ligne plus large à une extrémité. Ainsi, cette comète indique la direction et la tendance des vents. Le design de la comète avec sa tête et sa queue est écologique, car il correspond à notre perception de phénomènes naturels : les gouttes d'eau, les étoiles filantes ...

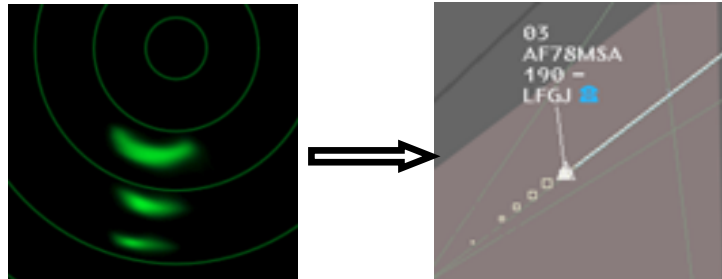


La comète radar visualisée par les contrôleurs utilise les propriétés visuelles de la comète introduite par Halley, mais ce design a été inspiré par des considérations technologiques. Les premiers écrans radars utilisaient des phosphores qui avaient une forte rémanence. Ainsi entre deux mises à jour, la position précédente de l'avion était encore visible avec une luminosité moindre. Le plot radar avait une durée de vie supérieure à la période de rafraîchissement du radar. La forme résultante indique la direction de l'avion, sa vitesse,



son accélération (différence des distances entre les plots), ainsi que sa tendance horizontale (virage à droite ou à gauche avec la courbure de la comète).

Le phénomène de rémanence dû à l'utilisation d'écran à phosphore a été conservé avec les nouvelles générations d'écrans cathodiques puis LCD. Ainsi, les positions passées sont visibles à l'aide de carrés dont la taille diminue avec l'ancienneté du plot radar (la plus ancienne position est représentée par le plus petit carré).



Ce design a été conservé avec les nouvelles version des visualisation radar car il possède des propriétés visuelles remarquables et permet de coder de nombreuses informations. Notre démarche de caractérisation va permettre de lister l'ensemble des informations disponibles et d'en comprendre la provenance.

Si on analyse de façon un peu plus précise le design de la comète, on constate que la forme résultante perçue par l'utilisateur est une ligne. La succession de carrés de tailles dégressives fusionne par l'effet de la continuité Gestalt [Kofka, 1935]. Cette fusion crée une ligne avec ses propriétés de longueur et de courbure. Le design et les informations codées par la comète radar sont résumés dans le tableau suivant. Toutes les données émergentes de la comète sont dues à la sémantique temporelle des positions de l'avion. Cette partie sera détaillée dans la partie concernant le design écologique.

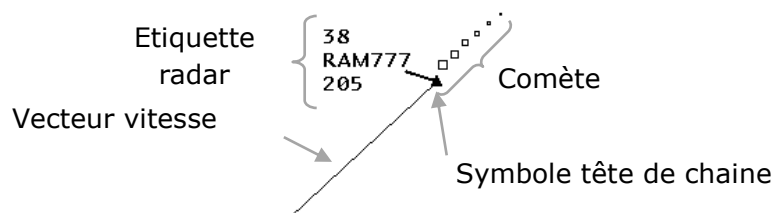
Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Point (Latitude, Longitude)	Longueur de la comète	Vitesse
	Courbure	Tendance
Taille(Temps)	Tête et queue	Direction
	Espacements des carrés	Accélération

## B. Description de quatre designs de comète

Dans cette partie, nous détaillons quatre designs utilisés dans le domaine de l'ATC pour afficher les positions des avions.

## 1. ODS: Le système de visualisation radar pour les contrôleurs aériens

La figure suivante rappelle les termes utilisés pour décrire la piste radar qui regroupe toutes les informations concernant un avion. Ce design est, comme nous l'avons vu précédemment, historique. En effet, la position la plus récente est donnée par le symbole tête de chaîne, et les données radar sont rafraichies toutes les quatre ou huit secondes en fonction du type de visualisation (4 secondes pour les approches, 8 secondes pour les centres en route). Il est à noter que ce taux de rafraichissement n'a rien à voir avec la vitesse de rotation des stations radar, mais est lié au temps mis par les systèmes annexes aux radars pour produire les positions calculées des avions.

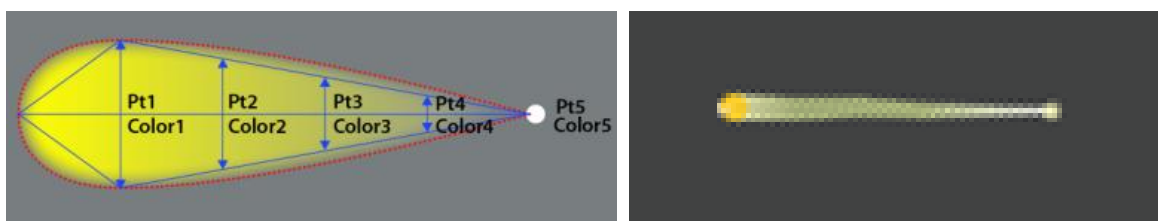


Le vecteur vitesse est une ligne qui part du symbole tête de chaîne et qui se termine à la position future de l'avion dans 3,6 ou 9 minutes (configurable par le contrôleur aérien). Ce vecteur vitesse est une extrapolation puisqu'il indique la position future de l'avion si celui-ci garde la même vitesse et la même direction cap. L'étiquette radar est composée de trois lignes. La première indique le niveau de vol de l'avion (son altitude en centaine de pieds, ft), la deuxième ligne contient l'indicatif de l'avion et la dernière ligne indique sa vitesse en nœuds (kts). Le leader est une ligne qui attache l'étiquette au symbole tête de chaîne.

Lors de la mise à jour de la piste radar (4 ou 8 secondes), les comètes radar « sautent » de leurs positions passées vers leur nouvelle position. Il n'y a pas d'animation interpolée pour lisser ces changements dans les données affichées.

## 2. RadarGL

RadarGL est un projet prospectif d'étude de nouvelles images radar. RadarGL affiche les positions courantes, passées et futures des avions avec des couleurs multiples, mais aussi de nombreuses entités visuelles, des menus avancés (Pie menu [Callahan et al., 1988], Marking menu [Kurtenbach & Buxton, 1991]) et des animations lissées. L'image suivante montre la comète radar telle qu'elle est affichée dans RadarGL.



Cette comète est une forme créée à partir des positions passées des avions. Elle est réalisée avec quatre points, chaque point ayant la même couleur, mais avec une transparence différente. Les points les plus récents sont opaques, les plus vieux sont transparents. Une bordure de deux pixels est ajoutée sur les bords de la comète pour

lisser son contour. Le dernier point de la comète se confond souvent avec le fond de l'écran (ceci est dû à sa transparence élevée), et rend difficile la perception de la longueur de la comète. Pour corriger ce problème, un point plus contrasté est ajouté à la queue de la comète.

Lors de la mise à jour des données radar, la comète est animée avec une interpolation linéaire entre sa position passée et sa nouvelle position. Il n'y a pas d'effet de saut dans la comète comme pour l'image d'ODS. Cette interpolation implique que le système n'affiche pas la dernière position connue de l'avion mais sa position avec dans le pire des cas 4 ou 8 secondes de retard. Il est à noter que dans ce cas c'est un choix de design qui impose cette contrainte.

La comète radarGL est avant tout un design issu des technologies avancées, il utilise les accélérations matérielles des cartes graphiques et permet ainsi d'utiliser des dégradés complexes, des animations...

### 3. ASTER

ASTER est aussi une visualisation prospective qui propose une vue verticale du trafic aérien contrairement à ODS et RadarGL qui proposent une vue de dessus. Comme nous l'avons déjà expliqué dans le chapitre sur la caractérisation des visualisations, dans Aster, les avions sont affichés dans un plan vertical : avec leur distance à un point de référence sur l'axe de X (abscisse) et leur altitude sur l'axe Y (ordonnée).

ASTER utilise aussi une comète pour afficher la position des avions avec deux designs différents.



La comète est créée à l'aide d'une forme dont le point Pt1 est placé à la position courante de l'avion. La longueur de cette forme est proportionnelle à la vitesse de l'avion et son orientation est liée à sa vitesse dans le plan verticale. La comète d'Aster permet aussi d'afficher la profondeur de l'avion par rapport à une droite de référence (IAF, balise de référence). La comète apparaît donc plus sombre et plus petite si l'avion est derrière le plan de projection, et plus gros et plus clair s'il est devant ce plan. Ce choix de design est écologique (plus prêt, plus visible) mais diminue le contraste avec la couleur du fond de l'image et ainsi dans certain cas la luminosité des la comète se confond avec celle du fond d'écran.

### C. Application du modèle de C&M sur les comètes

ODS, ASTER, radarGL affichent tous les positions des avions. Nous souhaitons comparer chacun de ces designs à l'aide du modèle de C&M.

Pour rappel, le tableau suivant énumère les légendes que nous utilisons pour la caractérisation des visualisations avec le modèle de C&M.

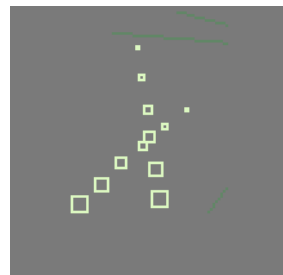
L	Ligne (line)
S	Taille (size)
Sh	Forme (shape)
F	Fonction
N, O, Q	Nominal, Ordonné, Quantitatif
Lon, Lat	Longitude, Latitude
Pt	Point
Orien	Orientation
T	Texte
Lum	Luminosité
C	Couleur, teinte

## 1. La comète ODS

La figure suivante montre une comète radar ODS d'un avion qui tourne à droite et qui a accéléré (irrégularité entre les points).



Nous avons déjà vu dans le chapitre précédent que les carrés qui forment la comète radar fusionnent pour former un seul objet (effet de continuité Gestalt [Kofka, 1935]). Ce choix de design permet aussi d'être dans une certaine mesure résistant au chevauchement ; l'utilisateur arrive à différencier les comètes même si elles se recouvrent.



Par caractériser la comète radar avec le modèle de C&M, nous avons adapté ce modèle pour faire apparaître le temps courant (Tcur) qui correspond à l'heure d'affichage de la donnée. La taille des carrés est linéairement proportionnelle à ce temps courant. Cette fonction agit comme un filtre en ne laissant visible (taille supérieure à 1 pixel) que les positions de l'avion qui correspondent à la comète. Le tableau suivant est la caractérisation de la comète ODS par le modèle de C&M, et les éléments en gris représentent les modifications que nous avons apportées au modèle initial :

Name	D	F	D'	X	Y	Z	T	R	-	[]	CP
X	QLon	f	QLon	P				Forme émergente			
Y	QLat	f	QLat		P						
T	Q	f(Tcur)	Q					S			

Cette caractérisation est incomplète car elle ne reflète pas toutes les informations perçues par les contrôleurs aériens. Par exemple la comète code la vitesse de l'avion avec sa longueur, la direction de l'avion avec sa courbure, son accélération avec la non régularité des distances entre points. Toutes ces informations sont très importantes pour les tâches de Contrôle. Le modèle de C&M ne peut pas refléter ces informations car il n'existe pas de données brutes pour les coder avec le modèle de Data Flow. On dit que ces informations sont *émérgentes*.

## 2. La comète ASTER

Le tableau suivant est l'application du modèle de C&M sur la comète ASTER. L'application du modèle peut se faire complètement dans C&M. Cette comète est une forme placée sur la position courante de l'avion. Cette forme à une taille qui est proportionnelle à la vitesse de l'avion et une orientation qui correspond à la vitesse verticale de l'avion.

Name	D	F	D'	X	Y	Z	T	R	-	[]	CP
Plot	Lat Lon (QxQ)	f	Q	P				Forme			
Altitude	Q	f	Q		P						
Vitesse verticale	Q	f	Q					O			
Vitesse horizontale	Q	f	Q					S			

## 3. Comparaison entre la comète ASTER et la comète ODS

Si on compare les deux tableaux issus de la caractérisation par le modèle de C&M de la comète ASTER et ODS, on peut remarquer que les informations codées par chaque design sont différentes. La comète ODS code la position de l'avion dans un plan Latitude/Longitude et « l'âge » de la position radar. La comète ASTER code la position de l'avion dans un plan Distance/Altitude, le taux de descente et la vitesse projetée de l'avion. Si on s'intéresse au nombre d'informations codées par chaque design, à l'aide de leur caractérisation, on voit que celle de la comète ASTER comporte quatre lignes et celle de la comète ODS trois lignes. Une conclusion naïve serait de dire que la comète ASTER code plus d'informations :

Informations codées par la comète ASTER	Informations codées par la comète ODS
Position de l'avion (Distance, Altitude)	Position de l'avion (Latitude, Longitude)
Vitesse de l'avion	Heure de chaque plot
Vitesse verticale de l'avion	

Comme précédemment expliqué, des informations émergent de la comète ODS comme la vitesse de l'avion, sa direction, son accélération. Nous avons rajouté ces informations émergentes en italique dans le tableau suivant.

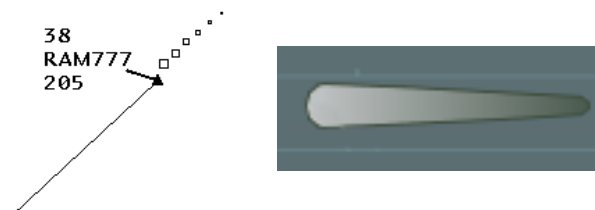
Informations codées par la comète ASTER	Informations codées par la comète ODS
Position de l'avion (Distance, Altitude)	Position de l'avion (Latitude, Longitude)
Vitesse de l'avion	Heure de chaque plot
Vitesse verticale de l'avion	<i>Vitesse de l'avion</i>
<i>Tendance (animation)</i>	<i>Tendance de l'avion (sa direction)</i>
	<i>L'accélération de l'avion</i>

Cette rectification du tableau permet de comparer objectivement les deux designs et ainsi de conclure que le design de la comète ODS code plus d'informations que le design de la comète ASTER. Il est à noter que nous avons ajouté une donnée émergente dans la comète ASTER. Il s'agit de l'émergence due à l'animation lorsqu'il y a une mise à jour de l'image. Si l'utilisateur perçoit cette mise à jour, il peut comparer la position précédente de la comète ASTER avec sa nouvelle position. Il peut alors en déduire la vitesse de modification des paramètres visuels du design et ainsi percevoir les vitesses d'évolution de l'information. Dans le cas de la comète ASTER, l'utilisateur pourrait donc percevoir l'évolution de la vitesse verticale de l'avion avec le changement d'orientation de la comète entre deux mises à jour ; le nombre d'informations codées serait le même qu'ODS. Cette information n'apparaît pas dans notre caractérisation car nous n'avons pas inclus de colonne de type animation. De plus, il est peu probable que cette information soit perçue par l'utilisateur car cette animation est trop courte dans le temps, voir inexistante.

#### 4. Comparaison entre la comète ASTER et le vecteur vitesse ODS

Pour rappel, le vecteur vitesse est une ligne qui part du symbole tête de chaîne et qui finit à la position future de l'avion si celui-ci garde la même vitesse et la même direction. Nous utilisons la notation de Bertin pour qui la ligne a une taille et une longueur. Dans le cas du vecteur vitesse, cette longueur est proportionnelle à la vitesse de l'avion. L'orientation de cette ligne est la même que celle de l'avion.

Si on compare visuellement les comètes ASTER et le vecteur vitesse d'ODS, on peut juste dire que les designs sont visuellement différents :



Par contre si l'on compare les tableaux issus de la caractérisation par le modèle de C&M de la comète ASTER et du vecteur vitesse ODS, on remarque une forte ressemblance.

Comète ASTER :

Name	D	F	D'	X	Y	Z	T	R	-	[]	C P
Point	Lat/Long (QxQ)	f	Q	P				Forme			
Altitude	Q	f	Q		P						
Vitesse	Q	f	Q					S			
Vitesse Verticale	Q	f	Q					O			

Vecteur vitesse ODS :

Name	D	F	D'	X	Y	Z	T	R	-	[]	CP
Longitude	Q	F	Q	L				Ligne			
Latitude	Q	F	Q		L						
Vitesse	Q	F	Q					S			
Direction	Q	F	Q					O			

Ainsi en comparant ces deux tableaux, on s'aperçoit que la comète ASTER ressemble beaucoup au vecteur vitesse d'ODS. En effet, même si la sémantique de position des formes n'est pas identique (dans un cas une vue du dessus en Latitude/Longitude et dans l'autre cas une vue verticale en Altitude/Distance), les informations codées et leur variable visuelle associée sont les mêmes. Les deux designs utilisent la taille pour coder la vitesse de l'avion et l'orientation pour la direction. En d'autres termes, on peut dire que la comète ASTER et le vecteur vitesse d'ODS sont identiques modulo une translation géométrique (le point de départ de la comète d'ODS est le point d'arrivée de la comète d'ASTER).

## 5. Conclusion sur la comparaison entre la comète ASTER et le vecteur vitesse ODS

Les designers utilisent le terme comète pour ASTER et nous avons montré que ce terme n'est pas approprié. La sémantique usuelle de la comète indique qu'elle contient les positions passées des avions, or la caractérisation par le modèle de C&M de la comète ASTER ne l'indique pas. De plus en comparant la caractérisation de la comète ASTER avec la caractérisation du vecteur vitesse d'ODS, nous avons constaté une forte ressemblance.

L'utilisation du terme comète peut donc être ambiguë : parle-t-on de la comète comme celle d'ODS ou comme celle d'ASTER ? Le problème peut être plus important si on essaie de retrouver la position passée de l'avion avec la comète ASTER. L'utilisateur peut alors en retirer de l'information erronée : contrairement à ODS, la queue de la comète ASTER ne correspond pas à la position passée de l'avion, mais indique seulement sa vitesse et son taux de descente.

Comme premier résultat, nous avons montré comment la caractérisation d'un élément graphique par le modèle de C&M peut être utile en soulevant un problème de sémantique : la comète ASTER n'est pas une comète au sens d'ODS. Ainsi, la

caractérisation donne aux designers un outil pour analyser, comparer et communiquer sur leur design.

## **D. Comparaison des comètes avec des critères qualitatifs**

---

Après avoir étudié, grâce au modèle de caractérisation de C&M, les données codées visuellement dans les comètes, nous proposons de compléter cette analyse avec d'autres critères. Nous souhaitons ajouter les informations dites émergentes, les propriétés visuelles des comètes et l'ensemble des spécificités qui ne peuvent pas être traduites avec le modèle de C&M. Notre démarche n'est pas formelle car elle n'utilise pas d'outil particulier. Pour faire l'analyse qui va suivre, nous avons itéré sur la caractérisation des designs en essayant de répondre à ces questions : « Quelles sont les informations perçues dans la comète ? », « Comment sont-elles perçues ? », « Pourquoi le designer de cette comète a fait ces choix de représentation ? ».

Le tableau suivant résume les caractéristiques visuelles des différentes comètes classées en trois catégories :

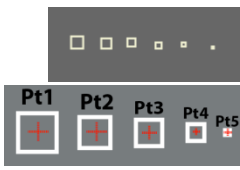
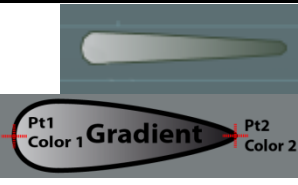
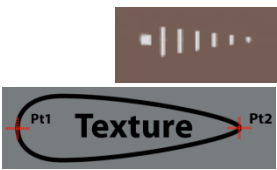
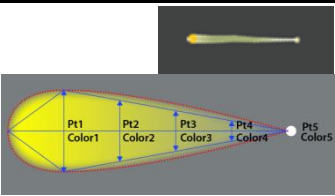
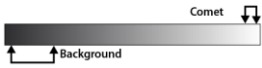
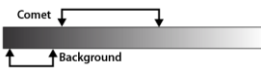
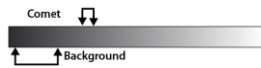
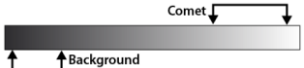
- Le design : cette catégorie concerne les choix de représentation.
- Les propriétés de design induites par les choix de représentation: ces propriétés sont souvent issues de la volonté du designer.
- Les informations disponibles : ce sont les informations que l'utilisateur peut percevoir.

Les trois catégories sont composées des propriétés suivantes :

- Le taux de rafraîchissement : il est *continu* si la position de l'avion est interpolée entre deux mises à jour des données radar. Il est *discret* si l'information est mise à jour en même temps que la mise à jour radar.
- L'invariance au zoom : indique les entités visuelles qui ne sont pas impactées par les modifications d'échelle.
- L'occlusion : indique si la comète est opaque ou transparente.
- La profondeur à l'écran : dans le cas de comètes se chevauchant, cette caractéristique indique comment est choisi l'ordre de chevauchement.
- Le contraste entre les luminosités des couleurs du fond de l'écran radar et les luminosités des couleurs de la comète. Si ce contraste n'est pas assez important, l'utilisateur risque de mal percevoir tout ou partie de la comète.
- La forme fixe : indique si toutes les comètes du design ont les mêmes formes.
- L'accélération : indique si l'accélération de l'avion est potentiellement perçue par l'utilisateur.
- La mort de piste radar : indique comment l'utilisateur est informé du non rafraîchissement d'une information radar dans le cas d'une perte de la piste d'un avion.



- La tendance : indique comment la tendance de l'avion est affichée. Cette tendance concerne les deux dimensions du plan l'affichage : Latitude/Longitude pour ODS et radarGl, Distance/Altitude pour ASTER.

Nom des Comètes	ODS	ASTER Design 1	ASTER Design 2	radarGL
Images des comètes				
<b>Design</b>				
Taux de rafraîchissement	Toutes les 4 à 8 secondes	Toutes les 4 à 8 secondes	Toutes les 4 à 8 secondes	Continue (fréquence l'affichage de l'écran).
Forme de la comète	Forme constituée de carrés de tailles différentes	Forme fixe qui peut changer de longueur	Texture	Forme variable
<b>Propriétés visuelles</b>				
Invariant au zoom	La taille des carrés formant la comète	Le dégradé de couleur dans la comète	Taille et forme de la texture	Forme et le gradient de la comète
Occlusion	L'intérieur des carrés est transparent	Comète opaque	La texture est partiellement opaque	Utilisation de la transparence dans le dégradé
Profondeur à l'écran	Non	Non	Non	Non
Résistance au recouvrement	Oui +	Oui ++	Oui +	Oui ++
Contraste avec le fond de l'écran				
Forme fixe	Non	Oui	Oui	non
<b>Données disponibles</b>				
Accélération	Affichée avec les distances entre les carrés qui forment la comète	Non	Non	Affichée dans avec le gradient de couleur qui remplit la comète
Profondeur en Z (altitude ou distance)	Non	Code la distance à une droite de référence par l'épaisseur de la comète.	Non	Non
Mort de piste radar	Disparition progressive des carrés	Disparition progressive avec de la transparence	Disparition progressive avec de la transparence	Arrêt du mouvement de la comète et disparition avec la transparence
Direction	Courbure de la forme de la comète	Orientation de la comète	Orientation de la comète	Courbure de la forme de la comète
Tendance	Courbure de la forme de la comète	Non, à moins de percevoir l'instant de la mise à jour des données	Non, à moins de percevoir l'instant de la mise à jour des données	Courbure de la forme de la comète
Vitesse	Longueur de la comète	Longueur de la comète	Longueur de la comète	Longueur de la comète
Position précise de l'avion	Oui	Non	Oui	Non
Dernière position de l'avion	Non	Non	Non	Oui
Positions radar passées	Oui	Non	Non	Oui

## 1. Utilisation du tableau des caractéristiques des comètes

Ce tableau permet de synthétiser les caractéristiques des comètes. Il peut ainsi être exploité par un designer qui chercherait à produire un certain type d'image radar et ainsi utiliser les propriétés d'une comète. Ce tableau peut aussi être utilisé pour améliorer les comètes existantes avec, par exemple, l'ajustement des luminosités pour augmenter les contrastes, ajouter un système d'ordonnement des comètes lorsqu'elles se chevauchent...

Ce tableau permet principalement de résumer de façon synthétique les caractéristiques de chaque comète.

## 2. Exploitation du tableau de synthèse des comètes

Nous avons confronté notre tableau avec les utilisateurs et les designers des comètes radar. De ces discussions a découlé un ensemble de conclusions et de constatations dont les principaux éléments sont les suivants :

- RadarGL et ASTER design 1 ont une meilleure résistance à l'occlusion. Cela signifie que lorsque plusieurs comètes se chevauchent, l'utilisateur peut plus facilement les différencier.
- Avec les designs d'ODS et de ASTER design 2, la comète est créée à partir de plusieurs entités visuelles. ODS utilise des carrés vides, et ASTER des barres verticales dans une texture. Cette texture est trompeuse dans le sens où elle induit l'utilisateur en erreur car par analogie avec la comète ODS, il peut penser que chaque barre verticale représente la position passée de l'avion alors que cette texture est étirée en fonction de la vitesse de l'avion. Cette remarque a été prise en compte dans les évolutions de design du logiciel et ainsi cette comète a été abandonnée au profit du design 1.
- La tendance de la comète (l'évolution de la direction de l'avion) ne peut être perçue avec le design d'ASTER (1 et 2) que si l'utilisateur perçoit la mise à jour des données radar, ce qui est peu probable et difficile (un changement toutes les 4 secondes). Pour se faire, l'utilisateur doit comparer les informations codées par la comète avant la mise à jour et en déduire les vitesses d'évolution. Avec le design d'ODS, cette tendance est toujours visible avec les positions passées de l'avion et la courbure de la comète.
- L'épaisseur de la comète d'ASTER n'est pas invariante au zoom alors que cette information code la profondeur de la position de l'avion par rapport au plan. Ce tableau a mis en évidence une erreur de conception qui n'avait pas été indiquée par les contrôleurs lors des évaluations du logiciel. Ceci est compréhensible, car les contrôleurs aériens n'ont pas à utiliser les fonctions de zoom: par défaut, toutes les informations disponibles sont affichées à l'écran avec un facteur de zoom suffisant. Nous avons découvert cette erreur de conception car nous ne nous sommes pas posés en tant qu'utilisateur du logiciel, mais comme analyste des propriétés visuelles.

## **E. Conclusion sur la caractérisation des comètes radar**

Notre contribution dans cette partie a été de montrer une utilisation de la caractérisation de C&M [Card & Mackinlay 1997]. Nous avons appliqué ce modèle à un certain nombre de représentations, nous l'avons amélioré, et nous l'avons utilisé [HURTER DSVIS 08], avec :

- la mise en évidence des ambiguïtés entre le design de la comète ASTER et ODS,
- la comparaison de designs au travers de leur description.

Nous avons aussi montré les limites de ce modèle de caractérisation en remarquant qu'il est impossible de caractériser complètement certains designs et en particulier les informations dites émergentes (design de la comète ODS).

Nous avons aussi souhaité pousser notre investigation sur ces designs en utilisant d'autres critères de caractérisation. Ils nous ont permis d'augmenter notre connaissance et notre compréhension des designs avec d'autres dimensions de caractérisation.

Nous avons donc proposé une méthode « ad hoc » qui peut être appliquée sur un ensemble de designs. Cette méthode n'est pas automatique, dans le sens où un analyste doit suivre des étapes propres à chaque design pour arriver à le caractériser complètement. Pour l'instant, notre méthode montre qu'elle peut donner des résultats mais n'est pas automatiquement applicable à d'autres domaines. Nous avons donc décrit une première approche pour arriver à caractériser des éléments graphiques de façon compacte et suffisamment précise pour être réutilisée par la suite par d'autres designers à la recherche d'un nouveau visuel.

Nos travaux permettent aux designers :

- d'évaluer les comètes radar et par extension d'évaluer d'autres designs,
- de réutiliser d'anciens designs de comète en comprenant comment les informations sont affichées et pourquoi les choix de design ont été réalisés,
- de communiquer entre designers avec un langage commun et précis,
- de formater les descriptions des designs.

### III. Design écologique et émergence

---

Les représentations graphiques qui cherchent à transmettre des informations utilisent les capacités perceptives visuelles des êtres humains afin de les aider à réfléchir efficacement [Card et al., 1999]. En particulier, les travaux issus de la visualisation d'information (InfoVis) ont engendré des représentations abstraites (Scatterplots Matrix) [Chambers et al., 1983], Conetree [Roberstion et al., 1991], échelles parallèles [Inselberg & Dimsdale, 1990], TableLens [Rao & Card, 1994], Tree-maps [Johnson & Shneiderman, 1991]... en s'appuyant sur des modèles de perception comme la sémiologie graphique [Bertin, 1983]. La sémiologie graphique ne prend pas en compte la sémantique des données. Seul le caractère des dimensions (Nominal, Ordonné ou Quantitatif) différencie les données. Ainsi, les « *timelines* » sont des représentations du temps utilisant une variable planaire cartésienne (le X ou le Y de l'écran). Dans ce cas, la perception des informations temporelles n'utilise que la capacité de perception de variables visuelles, ici une variable planaire.

Pourtant, la compréhension de la sémantique des données est importante pour faciliter leur interprétation. Dans son livre, Bertin [Bertin, 1983] ne s'y trompe pas et présente des représentations particulières pour une classe de données, celle des données spatiales (longitude, latitude). Les cartes, les plans sont une représentation « naturelle » ou « écologique » (termes que nous définirons plus loin) des données spatiales : on perçoit et on interprète le plan d'une maison, comme si on la voyait « du dessus ».

Dans cette partie, nous nous intéressons à des représentations « naturelles », ou « écologiques » des données temporelles, qui utilisent des phénomènes physiques caractérisant le temps. Sans le prouver, nous pensons que la sémantique apportée par ces représentations les rend plus faciles à interpréter que des représentations n'utilisant que des variables visuelles de la sémiologie graphique. Elles nécessitent moins de représentation explicite (date, axe temporel) et ainsi déchargent l'image. Comme elles sont interprétables directement [Gibson, 1977] et comme elles nécessitent moins de pixels [Tufte, 1983], elles permettent aux utilisateurs de réfléchir encore plus efficacement (dans le sens « *using vision to think* » [Card et al., 1999]).

Notre contribution est double. Tout d'abord, nous présentons un certain nombre de représentations écologiques de données temporelles, qui peuvent être réutilisées par les praticiens de l'IHM dans leurs propres interfaces. Puis, nous proposons une analyse systématique de ces représentations, afin de montrer que des informations qui ne sont pas présentes dans les données, émergent des représentations grâce à l'utilisation du temps. Cette analyse se fait par l'intermédiaire d'un modèle de caractérisation et d'un critère d'efficacité.

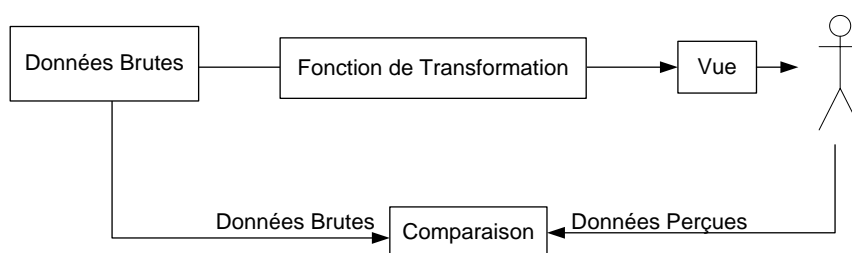
#### A. Définition de l'émergence

---

Quel que soit le modèle de Data Flow [Card et al., 1999] [Chi, 2000] utilisé, le principe de transformation des données vers un visuel reste le même. En entrée de la chaîne se trouvent les données brutes (par exemple les champs d'une base de données) et en sortie se trouve l'utilisateur qui perçoit ces données sous forme visuelle. Cette transformation utilise une Fonction de Transformation (FT) des données brutes vers la perception de l'utilisateur. En fonction des choix des variables visuelles et du design

utilisé dans la Fonction de Transformation, la perception de l'utilisateur peut être altérée ou augmentée.

Dans la majorité des cas, l'utilisateur interprète la donnée brute au travers de sa transformation visuelle, c'est-à-dire que pour être perçue, une donnée doit être transformée. Cependant, le processus de perception permet à un lecteur de percevoir plus d'informations que celles convoyées par les données transformées visuellement. Nous appelons données *émergentes* les données perçues par l'utilisateur sans pour autant avoir eu recours à la transformation visuelle d'une donnée brute. Par exemple si on considère l'affichage de la trajectoire d'un avion par une série de points, le design résultant est une ligne (groupement des points par effet « Gestalt » de la proximité et de l'alignement). La distance parcourue par l'avion est une donnée émergente dans le sens où le design utilise uniquement la position de l'avion pour afficher des points et non une donnée de la distance totale parcourue par celui-ci.



Le nombre de données émergentes est un critère qualitatif pour le choix de design. Le design le plus efficace est celui qui affiche le plus de données émergentes. Si on note DB les Données Brutes et DP les Données Perçues par l'utilisateur, on peut établir le critère qualitatif du design suivant :

- Si  $DB > DP$  le design est inefficace : il y a des données brutes transformées en représentation visuelles qui ne sont pas perçues par l'utilisateur.
- Si  $DB = DP$  le design est équilibré : il y a autant de données utilisées pour le design que de données perçues. En principe, ce sont les mêmes (chaque donnée brute est transformée visuellement et est perçue par l'utilisateur).
- Si  $DB < DP$  le design contient des données émergentes et est donc efficace.

## B. Illustration du phénomène d'émergence

---

Les représentations que nous présentons utilisent quatre phénomènes physiques se déroulant dans le temps : la disparition progressive, l'apparition progressive, le vieillissement, et le mouvement.

### 1. Représentations utilisant le phénomène de disparition progressive

Nous appelons la rémanence l'affichage de plusieurs représentations d'une même entité visuelle dans une seule image. Ce phénomène existe dans la nature, par exemple sous la forme de cercles dans l'eau qui disparaissent petit à petit, et qui permettent aux adeptes des ricochets de compter le nombre de rebonds.

### a) La comète radar

Les données émergentes de la comète radar ont déjà été identifiées dans les chapitres précédents. Le tableau suivant en est le récapitulatif :



Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Point (Latitude, Longitude)	Longueur de la comète	Vitesse
	Courbure	Tendance
Taille(Temps)	Tête et queue	Direction
	Espacements des carrés	Accélération

On peut remarquer que la comète code un nombre important d'informations émergentes : ce design est dans ce sens très efficace. Il a été utilisé sous plusieurs autres formes : *Mnemonic Rendering* [Bezerianos et al., 2006] toujours en utilisant la rémanence pour permettre à des utilisateurs de voir les changements non perçus. *Phosphor* a aussi utilisé la rémanence pour augmenter la puissance d'expression des widgets [Baudisch et al., 2006].

### b) Disparition de traces

MAMMI (Multi Actors Man Machine Interface) [Conversy et al., 2007] est un projet qui vise à instrumenter le travail collaboratif entre des contrôleurs aériens avec une table interactive multi-touch. Dans le prototype réalisé, les utilisateurs laissent une trace sur l'écran lors de chaque interaction. Par exemple le fait de glisser son doigt sur la table ou de déplacer un objet, laisse une trace qui va progressivement disparaître dans le temps. Cette propriété visuelle permet à un utilisateur de l'interface de comprendre les actions passées des autres collaborateurs. Sans cette trace, les autres utilisateurs seraient obligés de rester constamment en contact visuel avec l'interface pour comprendre l'enchaînement des actions. Grâce à l'utilisation d'une trace (qui est du même type que le design de la comète), ils disposent d'un historique, sur une période de temps définie, des positions passées des objets. Ce design utilise la perception écologique et est donc facilement compréhensible.



Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Ligne ( $X(t), Y(t)$ )	Courbure de la ligne, taille de la ligne.	Actions passées des utilisateurs

### c) Le vieillissement d'un objet

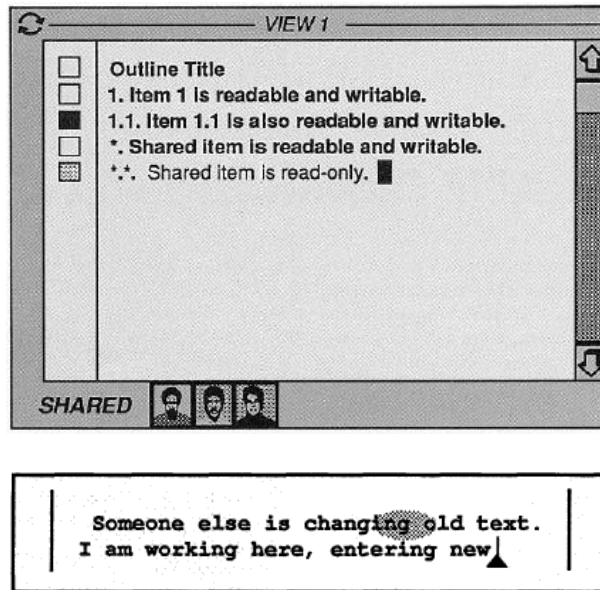


Le vieillissement est un phénomène naturel qui altère les propriétés visuelles des objets. Sa mise en œuvre dans des représentations permet à l'utilisateur de percevoir l'âge des objets. VidéoProbe [Conversy et al., 2003] est un dispositif qui permet de prendre des images de la vie familiale et de les partager entre plusieurs foyers d'une même famille. Les photos exposées subissent un vieillissement de leur apparence : elles restent affichées sans altération pendant un certain temps puis progressivement leur couleur disparaît, les contrastes s'estompent jusqu'à une disparition complète de l'image. Ainsi, l'âge de la photo est directement lié à son aspect visuel. C'est en ce sens que ce design est écologique : l'image vieillit comme pourrait vieillir une vraie photo, graphique, ou n'importe quel objet subissant l'usure du soleil.

GROVE [Ellis et al., 1990], un éditeur collaboratif, utilise un principe similaire. Les modifications effectuées par un utilisateur apparaissent dans la couleur qui lui a été assignée. Plus le temps passe et plus la couleur évolue vers la couleur par défaut du texte (le noir). Le texte ainsi ajouté subit un vieillissement et le lecteur peut très rapidement déduire l'ancienneté des modifications apportées. Cependant, à la différence de VideoProbe, la représentation n'est pas écologique, car elle n'imité pas un phénomène physique. Cette différence illustre l'intérêt de tenir compte des phénomènes perceptifs de plus haut niveau. Il semble en effet indispensable de refléter cette différence de



caractère de perception (que nous estimons plus efficace) lors d'une comparaison entre visualisations.



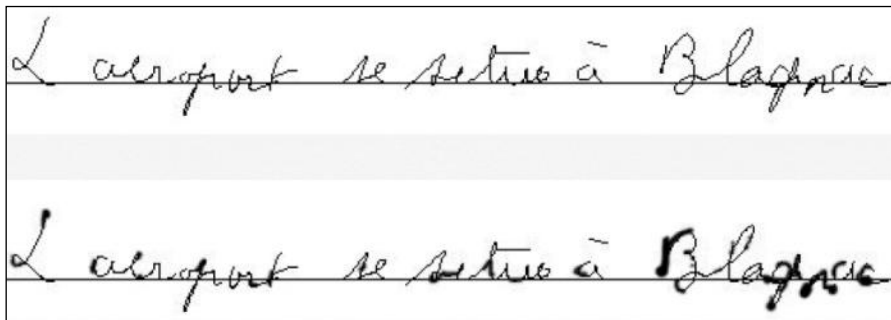
## 2. Représentation utilisant le phénomène d'accumulation

L'accumulation, comme par exemple celle de la neige qui tombe, est un phénomène naturel : on peut inférer le temps qui s'est écoulé par la perception de l'amoncèlement.

### a) Le papier buvard

La représentation suivante vise à vérifier les effets d'un médicament sur les patients atteints de la maladie de Parkinson. Pour ce faire, le patient réalise deux tâches d'écriture à la main : une avant la prise du médicament, l'autre après. Si le médicament est efficace, la deuxième écriture devrait être plus régulière, notamment dans la dynamique du tracé.

La représentation classique qui utilise des graphes de vitesse en X et en Y, est difficilement interprétable et ne permet pas d'évaluer la qualité de l'écriture. Nous avons donc utilisé la métaphore du papier buvard. Le papier buvard absorbe l'encre d'un stylo en la diffusant sur la surface du papier. Plus longtemps le stylo reste en contact avec le papier, plus la tâche d'encre est grande. Lors du déplacement d'un stylo, l'épaisseur du trait est donc directement proportionnelle à la vitesse de déplacement du stylo sur le papier.



Ce design illustre comment une image par définition statique peut coder des informations qui ont évolué dans le temps (ici la variation de la vitesse). Ce design est écologique car un analyste peut rapidement reconstruire les mouvements exécutés pour réaliser l'écriture. Des spécialistes ont ainsi pu déterminer les directions bloquantes pour les patients, ou les zones d'arrêts psychomoteurs.

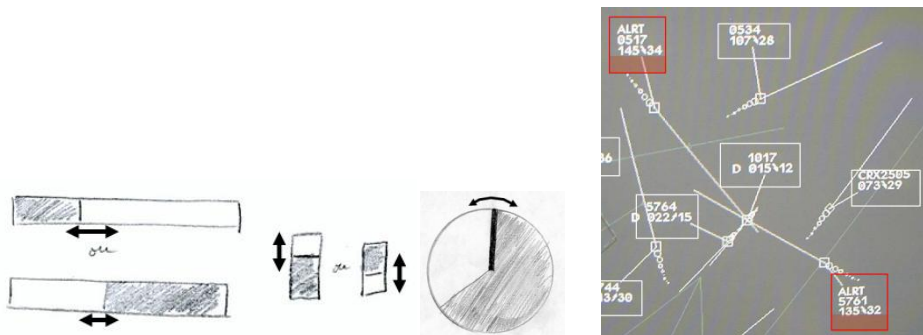
Les données émergentes sont la qualité de la courbe, la vitesse globale de l'écriture, mais aussi l'évolution de la maladie dans le temps (comparaisons de plusieurs traces).

Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Ligne ( X(t), Y(t) )  Taille (V)	Forme du tracé	Qualité de la trace
	Forme du tracé	Vitesse globale de déplacement
	Patterns	Lettre bloquantes
	Patterns	Direction de prédilection

Il est à noter que la perception temporelle du déplacement sur la trace est uniquement due à l'utilisation de l'épaisseur de la ligne. Et c'est cette épaisseur qui fait émerger de l'information.

### b) Alarmes jauges, accumulateurs

Les systèmes radar utilisés par les contrôleurs aériens sont capables d'afficher une alarme lorsque deux avions sont trop proches. Elle se déclenche deux minutes avant le passage sous la norme de séparation, et se traduit visuellement par un texte clignotant : « ALRT ». Cette alarme est un indicateur d'état : il y a ou il n'y a pas d'alarme. Ce design visuel ne répond pas aux questions suivantes: « dans combien de temps va avoir lieu l'alarme ? » ou « depuis combien de temps l'alarme est activée ? ». Des travaux ont été réalisés pour étudier des solutions alternatives à la représentation d'alarmes temporelles. La recherche de ces solutions a consisté à effectuer un ensemble de brainstormings avec des designers et des contrôleurs aériens. Des études ont permis de concevoir de nouvelles visualisations [Athènes et al. 2000]. La solution envisagée est d'utiliser des jauges et des accumulateurs en forme de rectangles ou de cercles.



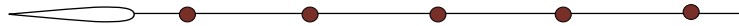
Ce design écologique permet de comprendre rapidement si l'alarme vient juste d'apparaître (la jauge est vide) ou si la situation est urgente (la jauge est pleine). Les

états de remplissage de la jauge sont quantitatifs ce qui permet d’avoir une gradation assez fine sur la temporalité d’une alarme.

Fonctions de transformation	Codes visuels source de l’émergence	Données émergentes
Surface ( temps avant alarme )	Remplissage d’une jauge	Criticité de l’alarme
	Remplissage des jauges	Ordonnancement des alarmes

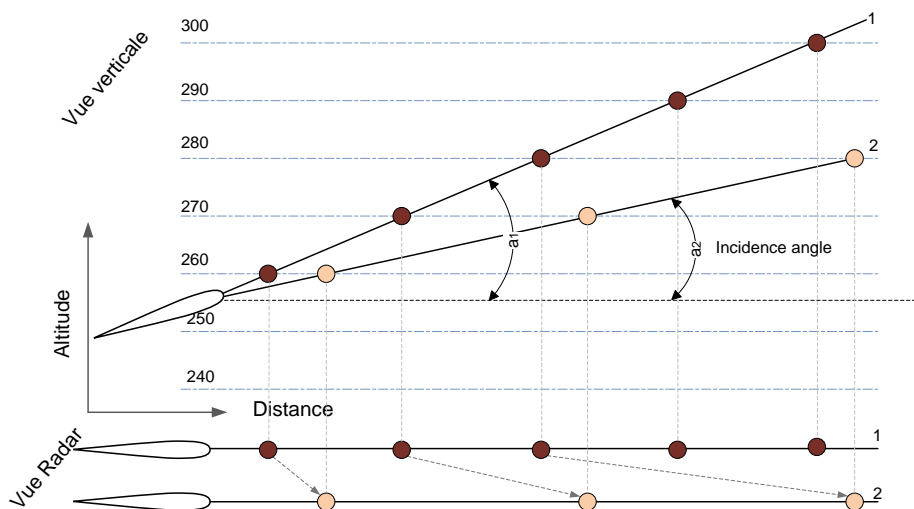
### C. Représentation utilisant le mouvement

Le Passage de Niveau de Vol (PNV) est un design qui consiste à afficher devant la comète les positions futures de l’avion à son passage à une altitude multiple de 10. Pour ne pas surcharger la visualisation, l’indication d’altitude n’est pas donnée. Seule l’altitude courante de l’avion est indiquée dans l’étiquette radar.

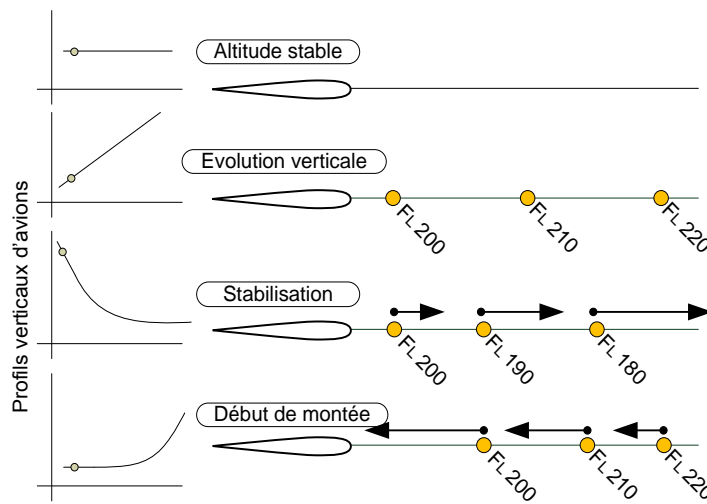


Par exemple dans l’image suivante, l’avion va croiser le niveau de vol (altitude) 260 puis 270, 280... Chaque passage de ces altitudes est symbolisé par un point. En comparant la position des points du PNV de deux avions, le contrôleur peut rapidement vérifier si un vol en évolution verticale (qui monte ou qui descend) peut générer un conflit avec un autre vol. Dans la figure suivante, deux profils verticaux sont explicités. Le PNV résultant permet avec la vue de dessus (bas de la figure) de percevoir rapidement si l’avion monte vite ou lentement : par exemple l’espacement des points indique que l’avion 1 monte plus vite que l’avion 2.

La partie haute du dessin représente le profil de vol vertical de l’avion. La partie basse de l’image représente l’image affichée au contrôleur aérien. Des lignes verticales ont été rajoutées pour comprendre le processus de création de l’image, elles permettent de faire la correspondance entre la position des points et le passage de l’avion à un niveau de vol.



Cette représentation fait partie d'une étude prospective, et n'est pas utilisée en situation opérationnelle. L'image statique permet de faire émerger la vitesse verticale de l'avion et son accélération mais elle n'est pas écologique. Cependant, si un changement de profil vertical est accompagné d'une transition animée, la perception devient écologique : c'est la métaphore d'un objet longiligne (comme un bâton), que l'on couche ou que l'on relève en le regardant du dessus. S'il est proche d'une position verticale, on ne voit qu'un point ; si on le couche, sa projection dans le plan horizontal « s'étale ». La figure suivante donne quatre exemples de profils verticaux d'avions qui évoluent dans le temps. Ainsi le PNV fait évoluer ses points : ils s'étalent jusqu'à leur disparition lors d'une stabilisation de l'avion sur une altitude, et ils se tassent lors de la mise en montée ou en descente de l'avion.



C'est la dynamique de l'image qui permet de faire émerger de l'information du design : le début et la fin de l'évolution de l'avion.

Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Point [ Latitude (temps), Longitude (temps), Altitude (temps) ]	Espace entre points	Valeur absolue de la Vitesse verticale
	Espace entre points	Accélération verticale
	Début/fin de l'animation	Début/ fin d'évolution

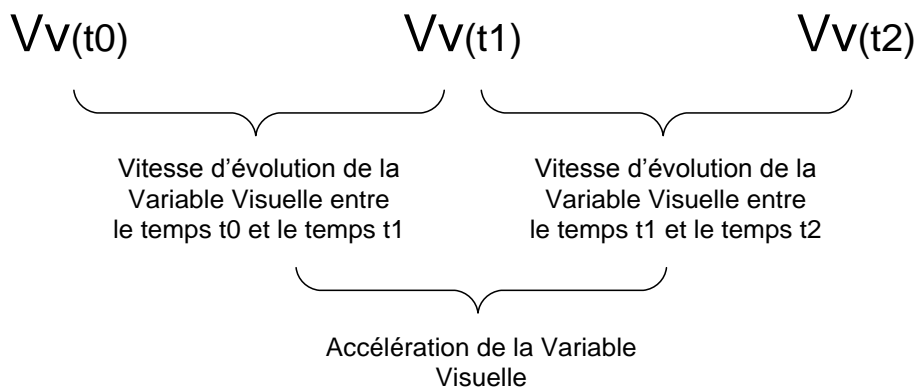
Les systèmes actuels utilisés par les contrôleurs aériens affichent les informations d'évolution verticale des avions sous forme de texte dans l'étiquette radar. La tendance de l'avion est donnée par une flèche pour indiquer la direction et par une valeur textuelle. L'étiquette radar ne permet pas de comparer les profils verticaux des avions mais seulement leur vitesse verticale instantanée. De plus l'étiquette radar ne donne pas de notion temporelle concernant l'évolution du profil vertical. Cette comparaison entre l'étiquette radar et le PNV permet de mettre en avant l'intérêt des données émergentes de ce design.

## D. Discussion sur l'utilisation de la perception écologique

Dans cette partie, nous expliquons pourquoi et comment la perception écologique des designs affichant des données temporelles crée de l'émergence. Cette émergence trouve ses sources dans la rémanence avec par exemple le design de la comète, dans l'accumulation avec des jauges, ou dans le mouvement avec le design du passage de niveau.

### 1. L'émergence de la vitesse d'évolution d'une variable visuelle

Les designs qui utilisent la rémanence affichent plusieurs occurrences d'une même donnée à des temps différents. Ainsi le lecteur va pouvoir faire des comparaisons entre ces occurrences. Dans le cas de la comète radar, plusieurs occurrences du même avion sont affichées à des instants différents (utilisation de la Variable Visuelle « position »). L'utilisateur peut alors comparer la position sur l'écran de deux occurrences à deux instants différents et ainsi évaluer une vitesse d'évolution. Par extension, il pourra évaluer l'accélération avec au moins trois occurrences. Cette perception (et l'émergence) de la vitesse et de l'accélération peut se faire avec les variables visuelles Ordonnées (O) ou Quantitatives (Q) : taille, luminance, saturation, position, orientation.



Le cas de la comète radar est un peu à part car ce design utilise plusieurs variables visuelles qui évoluent avec le temps. La Variable Visuelle position est utilisée pour afficher la position de l'avion à plusieurs instants. Cette rémanence géographique permet de percevoir la vitesse de l'avion, son accélération et ainsi le taux de virage de l'avion. La comète utilise aussi la taille pour coder l'âge des plots radar. Cette deuxième forme de la rémanence permet de coder la vitesse de vieillissement des plots radar. Dans le cas du Contrôle Aérien, cette vitesse est constante pour tous les plots radar (la mise à jour des informations est liée au taux de rafraîchissement des radars). La dégression régulière de la taille des plots radar permet de les grouper par effet Gestalt. Ainsi, la forme résultante permet de déterminer la direction de l'avion.

## 2. L'émergence due à la dynamique de l'image

Lorsque la dynamique de l'image est utilisée pour coder des informations temporelles, c'est la mémoire de l'utilisateur qui est mise à contribution. La perception de la vitesse et de l'accélération d'une variable visuelle par l'utilisateur est quasiment inchangée (comparaison des évolutions des Variables Visuelles). Cependant, l'utilisateur doit créer une image mentale pour évaluer la modification des variables visuelles entre deux instants. Ce choix de design est moins précis que l'utilisation d'une image statique car il existe des incertitudes sur les intervalles de temps. L'être humain se trompe dans son estimation des intervalles temporels : il a tendance à sous-estimer les petits intervalles de temps et à surestimer les grands (loi de Vierordt [Lejeune & Wearden, 2009]). L'Homme se trompe aussi sur la mémorisation de l'état des variables visuelles (la précision de l'image mentale décroît en fonction du temps).

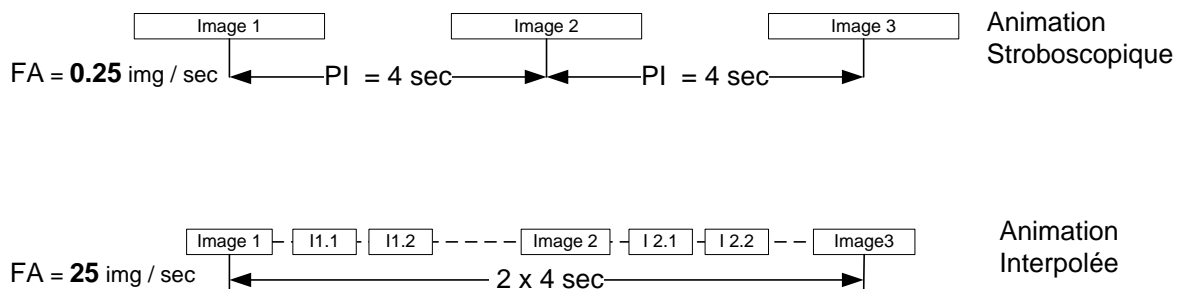
## 3. L'émergence due à l'accumulation

Le design de l'accumulation utilise le même principe de comparaison des valeurs des variables visuelles. Dans ce cas on ne parle pas de vitesse et d'accélération d'une variable visuelle (au sens de Bertin), mais plutôt d'une vitesse et d'une accélération perçues du remplissage de la jauge. Cette vitesse et cette accélération émergent.

Ce design a des propriétés supplémentaires ; on a une estimation assez précise du temps passé (la partie remplie de la jauge) et du temps restant (la partie à remplir de la jauge). En revanche, il n'y a pas d'information sur l'historique des positions passées et pas d'information sur les positions futures.

## E. L'émergence due à l'animation

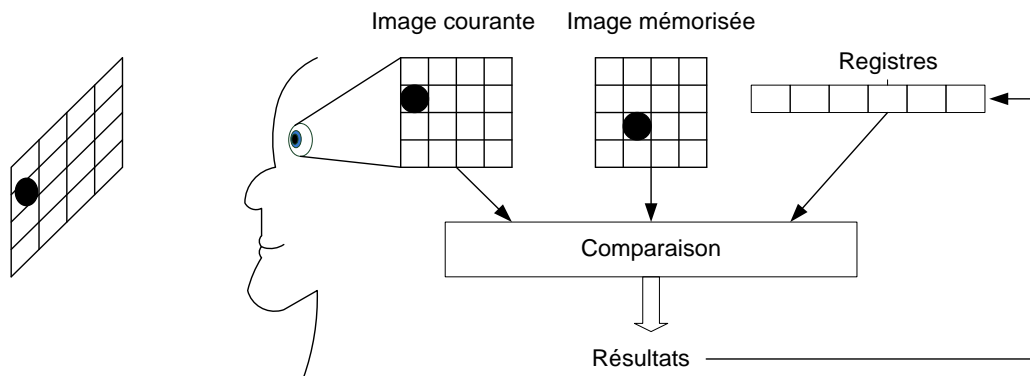
Comme précédemment expliqué, l'animation peut coder de l'information. Par exemple, le Passage de Niveau de Vol code, grâce à l'animation, l'évolution verticale de l'avion. Dans cette partie, nous allons expliquer pourquoi certains designs permettent de coder de l'information avec l'animation de leur visuel. Pour ce faire, nous utilisons un modèle simplifié de la perception de l'animation par l'être humain. Une animation est composée d'une succession d'images qui se substituent dans le temps. Chaque image est affichée pendant un laps de temps appelé « Période de l'Image » (PI). La vitesse de défilement des images est calculée avec le rapport  $1/PI$  et est appelé la Fréquence d'Affichage (FA). Si la période entre deux images est supérieure à 60 ms (ce qui correspond à 15 images par secondes), l'utilisateur perçoit des saccades dans l'animation. Pour supprimer cette perception et obtenir une animation fluide, il est possible de créer des images intermédiaires en utilisant l'interpolation.



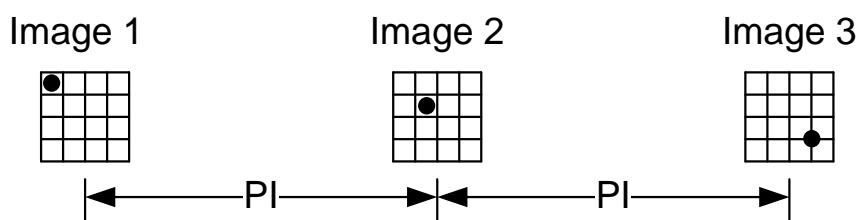
## 1. Modèle de perception de l'animation

Pour expliquer la transmission d'informations avec l'animation, nous avons élaboré un modèle épuré de la perception. Ce modèle est une vision simpliste mais fonctionnelle du processus de perception de l'animation par la comparaison d'images successives.

L'utilisateur perçoit les images et les mémorise dans un espace mémoire [Card et al., 1986]. Card, Moran et Newel ont détaillé ce processus avec des métriques validées par des expérimentations. Par exemple, l'utilisateur garde en mémoire les informations de l'image entre 90 ms et 1000 ms. L'utilisateur peut ensuite comparer les informations mémorisées avec l'image courante affichée et ainsi percevoir et évaluer les différences. Bien sûr, cette comparaison est d'autant plus imprécise que les images sont complexes et que le temps d'affichage entre les deux images est long. Le résultat de cette comparaison peut être mémorisé dans des registres pour une utilisation future [Miller, 1956]. Ce modèle permet de comprendre certaines limitations physiologiques ; si le temps pris pour réaliser l'opération de comparaison est supérieur au temps de mémorisation d'une image, la comparaison n'est plus réalisable.



Pour comprendre ce modèle, nous allons prendre l'exemple d'un point qui se déplace. L'utilisateur va voir une succession de trois images qui contiennent chacune un point à une position différente.



Notre modèle permet d'expliquer comment l'utilisateur va percevoir un point en mouvement qui accélère. Cette perception du mouvement est basée sur l'évaluation de la distance de déplacement du point entre deux images. Cette distance est donnée par la formule suivante :

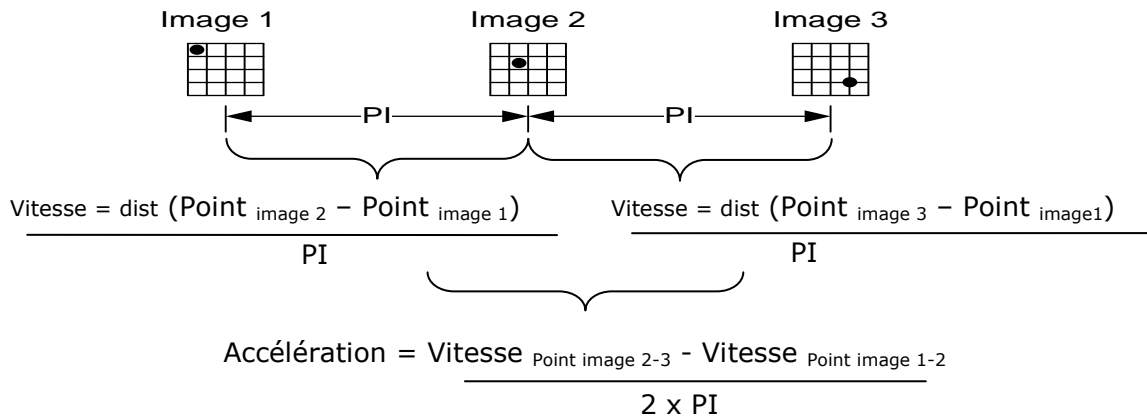
$$\text{Dist} = \text{distance} (\text{Position}_{\text{point image 2}} - \text{Position}_{\text{point image 1}})$$

Sachant que l'image à une fréquence de rafraîchissement fixe, on peut calculer une vitesse de déplacement du point :

$$\text{Vitesse}_{\text{point image 1-2}} = \text{Dist} / \text{PI}$$

Et parce que PI (la période de l'image) est constante, l'utilisateur peut la comparer avec la vitesse de déplacement du point entre les 2 images suivantes (Vitesse<sub>point image 2-3</sub>).

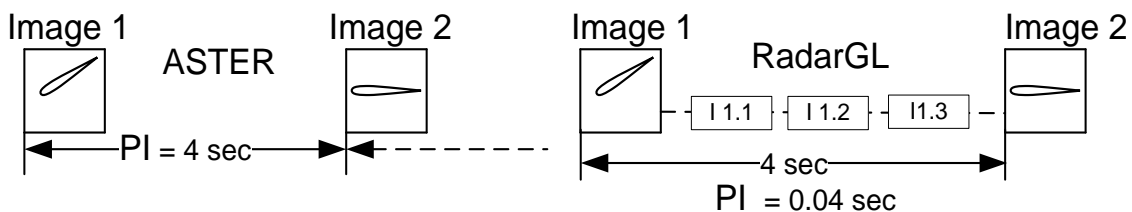
Le fait que PI soit constantes a son importance car si ce n'était pas le cas, la vitesse de déplacement du point entre deux images consécutives ne pourrait pas être comparée, et donc l'accélération du point ne pourrait pas être codée. Le schéma suivant résume comment la vitesse et l'accélération du déplacement du point est codée par l'animation de l'image :



Cet exemple utilise la variable visuelle de position du point pour créer l'animation, mais le même principe peut être appliqué avec les autres Variables Visuelles : taille, couleur, luminosité, saturation, orientation.

## 2. Animation avec interpolation des Variables Visuelles entre deux images

La fréquence de rafraîchissement des données radar est constante (4 secondes). Prenons le cas où, entre les images 1 et 2, l'orientation de la comète change. La différence d'orientation entre les images 1 et 2 permet à l'utilisateur de percevoir la vitesse d'évolution de l'orientation de la forme. Avec la comète ASTER, l'animation saute entre l'orientation initiale de la comète affichée par l'image 1 et l'orientation finale affichée par l'image 2.



Pour la comète radarGL, l'animation est réalisée avec de nombreuses images intermédiaires, ce qui crée une progression lissée de l'orientation de la comète.

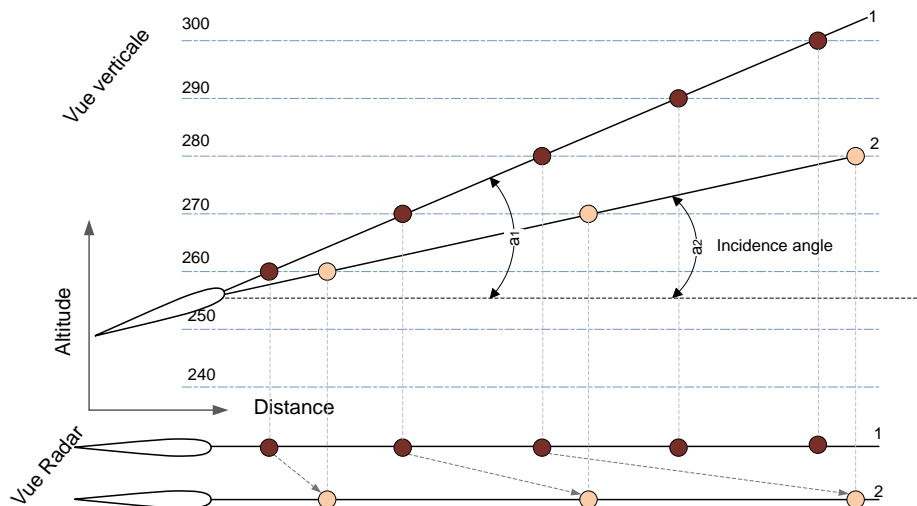
Une première constatation peut être déduite en comparant ces deux designs d'animation : le temps disponible pour percevoir que la comète change d'orientation est bien moins important avec la comète ASTER qu'avec la comète radarGL. En effet, pour la comète ASTER, l'utilisateur doit dans le pire des cas attendre PI (4 secondes) pour voir la comète changer d'orientation, alors qu'avec radarGL, l'utilisateur doit attendre PI (0.04 seconde) au maximum pour percevoir la même information. Avec ces considérations,



nous pouvons conclure que l'animation avec l'interpolation est meilleure pour améliorer la transmission d'informations.

### 3. Analyse de l'animation du passage de niveau

Pour comprendre les informations codées par le passage de niveau (PNV), nous reprenons l'animation codant un changement de niveau de vol d'un avion. L'avion reste en montée mais va diminuer sa vitesse verticale en passant de la pente notée 1 à la pente notée 2. L'animation résultante est un étalement des points le long du vecteur vitesse de l'avion :



Nous proposons de faire une analyse de ce design en utilisant la géométrie cartésienne.

La trajectoire de l'avion dans le plan vertical est définie par sa position initiale (altitude<sub>t0</sub> , Distance<sub>t0</sub>), sa vitesse initiale V<sub>t0</sub>, et l'incidence de l'avion i<sub>t0</sub>.

Le vecteur vitesse de l'avion a deux composantes dans le plan cartésien:

$$\begin{cases} V_x = V \times \cos(i) \\ V_y = V \times \sin(i) \end{cases}$$

L'équation paramétrique de la trajectoire de l'avion, en fonction du temps t, est donc la suivante :

$$\begin{cases} Altitude = t \times V_x + Altitude_{t0} \\ Distance = t \times V_y + Distance_{t0} \end{cases} \text{ avec } t \in [0, +\infty[$$

$$\begin{cases} t = \frac{Altitude - Altitude_{t0}}{V_x} \\ Distance = \frac{(Altitude - Altitude_{t0}) \times V_y}{V_x} + Distance_{t0} \end{cases} \text{ avec } Altitude \in [0, 10, 20... 390,400]$$

$$Distance = \frac{(Altitude - Altitude_{t0}) \times \Psi \times \sin(i)}{\Psi \times \cos(i)} + Distance_{t0}$$

$$Distance = \frac{(Altitude - Altitude_{t_0})}{\tan(i)} + Distance_{t_0}$$

Ce dernier résultat indique que pour une altitude donnée, la position du point de passage de l'avion à cette altitude donnée, ne dépend que de son incidence:  $\tan(i)$  est la seule valeur non constante de l'équation. Ainsi, pour un avion qui garde son profil de vol constant (pas de changement d'angle d'incidence de l'avion), la position des points sur le vecteur vitesse ne bouge pas, même si l'avion avance ! Par contre si l'incidence de l'avion évolue dans le temps, l'équation précédente indique que la distance entre l'avion et chaque point du vecteur vitesse va changer. Ce qui correspond au phénomène observé avec ce design.

Il est intéressant de remarquer que le but initial du design lors de sa création était de coder visuellement la position future de l'avion à une altitude donnée. Avec notre analyse analytique du design, on comprend que l'animation va coder une information supplémentaire : l'évolution de l'incidence de l'avion.

#### **4. Le designer face à l'émergence**

Les designers cherchent à créer des visuels efficaces. Les représentations écologiques sont intéressantes car elles permettent aux utilisateurs de percevoir de l'information en minimisant leur charge cognitive. Ce phénomène est parfois dû au hasard car il n'existe pas de méthode ni de théorie indiquant si un design va faire émerger de l'information avec une perception écologique ou non écologique. Seule l'évaluation empirique avec des utilisateurs permet de déterminer l'existence de telles caractéristiques.

Par exemple, le design du passage de niveau n'avait pas initialement été conçu pour faire émerger le début et la fin d'évolution de l'incidence de l'avion (cette émergence est due à la dynamique de l'image).

Cependant, il existe des cas où l'émergence n'est pas souhaitée. L'émergence est souvent liée à la perception écologique. Ainsi l'utilisateur perçoit de l'information écologique automatiquement. Si on prend le cas d'une barre de progression (rectangle qui se remplit progressivement dans le temps), l'utilisateur va percevoir la vitesse et peut-être son accélération de remplissage. Si cette vitesse ne code pas volontairement de l'information, l'utilisateur va de toute façon percevoir cette donnée qui risque de l'induire en erreur. Par exemple si la barre ne bouge pas, l'utilisateur peut penser que le système est bloqué, alors que la barre de progression indique le nombre d'éléments en cours d'analyse, et non le temps prévu pour analyser l'ensemble des informations.

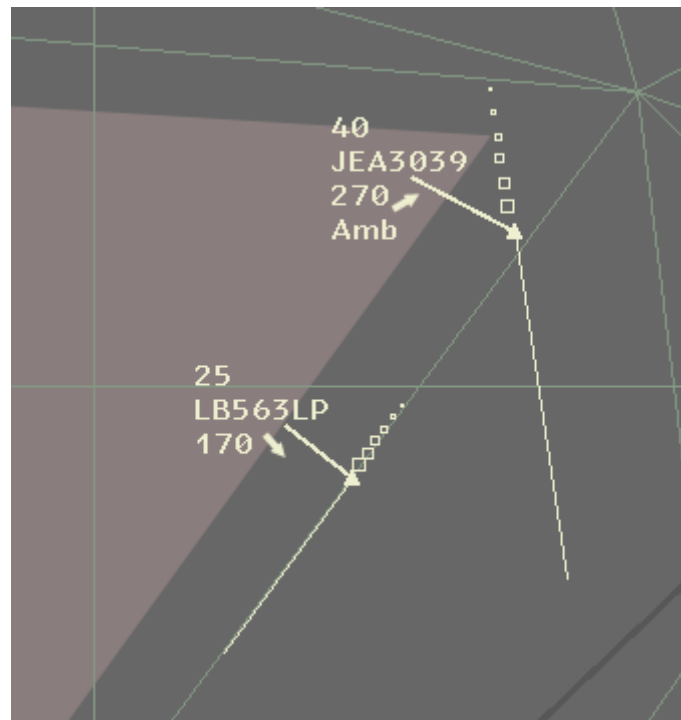
#### **F. Exemple d'utilisation du design écologique**

---

Dans cette partie, nous proposons d'utiliser notre caractérisation pour améliorer le design du PNV. Ce design est très efficace car il transmet beaucoup d'informations et aide ainsi les contrôleurs aériens à mieux comprendre le profil vertical des avions. Mais en analysant les informations transmises par ce design, le tableau suivant indique que seule la valeur absolue de l'évolution verticale de l'avion est représentée. Ce design seul ne permet pas d'indiquer au contrôleur aérien si l'avion monte ou s'il descend.

Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Point [ Latitude (temps), Longitude (temps), altitude (temps) ]	Espace entre points	Valeur absolue de la Vitesse verticale
	Espace entre points	Accélération verticale
	Début/fin de l'animation	Début/ fin d'évolution

Pour pallier la déficience de ce design, une flèche est accolée à l'étiquette radar pour indiquer si l'avion monte ou descend. Mais cette flèche oblige les contrôleurs à changer leur point d'attention de la comète vers l'étiquette radar.

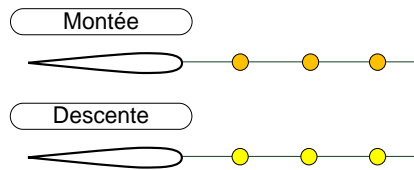


Dans cette partie, nous proposons d'utiliser notre modèle de caractérisation pour trouver un nouveau design permettant de coder la direction de l'évolution verticale des avions en utilisant des données émergentes. Pour ce faire, le designer peut utiliser une des Variables Visuelles disponibles : la taille, la couleur, la forme, la texture...

### 1. Utilisation de la couleur pour coder la direction verticale des avions

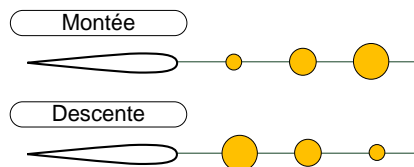
Le choix initial des designers pour coder la direction verticale des avions s'est porté sur l'utilisation de la couleur. Quand les points sont jaunes, l'avion monte. Quand les points sont oranges, l'avion descend. Ce choix de design n'est pas optimal pour au moins deux raisons : premièrement, la couleur jaune est perceptuellement proche de la couleur orange (même teinte) ce qui peut conduire à des erreurs d'interprétation en cas de stress des contrôleurs. Deuxièmement, l'utilisation de couleurs arbitraires oblige les utilisateurs

à faire la correspondance couleur/signe de la vitesse verticale et nécessite donc une période d'entraînement.



## 2. Utilisation de la taille pour coder la direction verticale des avions

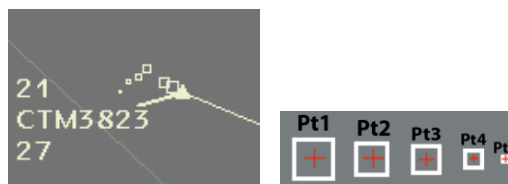
Un autre choix de design consiste à utiliser la taille de chaque point pour indiquer la direction verticale des avions. Dans ce design, la taille de chaque point est directement proportionnelle à l'altitude de l'avion à la position du point. Les points les plus gros correspondent à l'altitude la plus élevée, alors que les points les plus petits correspondent à l'altitude la plus faible.



Ce design permet de faire émerger de l'information : les utilisateurs perçoivent directement l'information de profil des avions avec une visualisation de dessus. Plus les objets sont haut, plus ils sont gros car « proches » du point de vue de l'utilisateur.

Fonction de transformation	Codes visuels source de l'émergence	Données émergentes
Point (latitude, longitude)	Longueur de la comète	Vitesse
	Progression régulière de la taille	Direction verticale
Taille (altitude)	Espacement entre les formes	Accélération

Malheureusement, ce design s'oppose au design des comètes radar qui utilisent les mêmes propriétés visuelles : l'utilisation de formes proches avec une régularité des modifications de tailles.

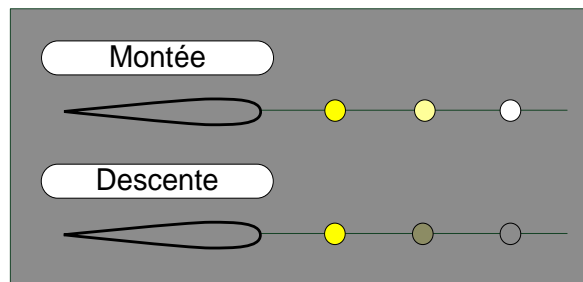


Fonctions de transformation	Codes visuels source de l'émergence	Données émergentes
Point (Latitude, Longitude)  Taille(Temps)	Longueur de la comète	Vitesse

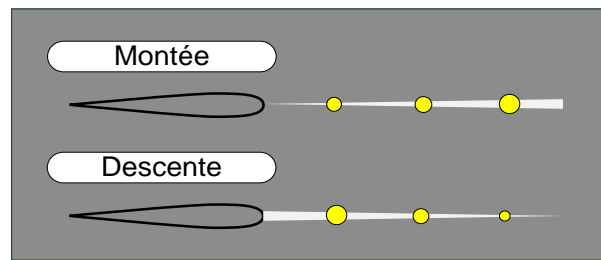
La comète radar permet de coder la direction des avions dans le plan horizontal alors que ce nouveau design permet de coder la direction des avions dans le plan vertical. Ceci pose un problème de cohérence pour les contrôleurs aériens qui utilisent régulièrement les comètes radar et pourrait aussi entraîner de mauvaises interprétations de données.

### 3. Utilisation d'un design écologique pour coder la direction verticale des avions

Les designers ont utilisé deux autres designs dits écologiques. Le premier utilise l'effet naturel de la saturation des couleurs : plus les objets sont loins moins ils sont saturés (leur couleur est grisâtre), plus les objets sont proches, plus ils sont saturés (avec des teintes prononcées). Les points du design écologique sont ainsi saturés si l'altitude est élevée (proche de l'utilisateur en vue de dessus) et dé-saturés si l'altitude est faible (loin de l'utilisateur).



Le deuxième design écologique utilise la notion de rampe. Les points sont ainsi positionnés sur une rampe vue de dessus. Ainsi la largeur de la rampe est importante pour les hautes altitudes et fine pour les basses altitudes. Pour éviter que la modification de la taille des points ne ressemble au design de la comète, le designer utilise une altération de taille de quelques pixels. Cette altération est suffisante car elle est renforcée par le design de la rampe pour percevoir la notion de profondeur.



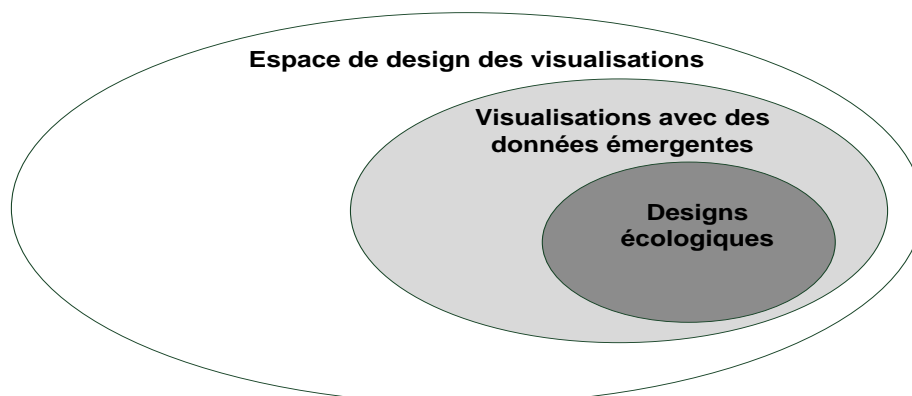
## G. Conclusion sur l'utilisation de l'écologie

---

Dans cette partie, nous avons étudié des représentations écologiques, c'est-à-dire des représentations qui se servent des connaissances acquises par l'Homme dans son environnement pour transmettre de l'information. Nous avons utilisé une méthode ad-hoc pour expliciter la bande passante entre les informations disponibles dans les designs et leur perception par les utilisateurs. Avec un tableau qui synthétise la description des designs, nous avons défini et inclus les informations émergentes. Nous avons discuté des sources de l'émergence et nous avons défini un critère d'évaluation des visualisations avec un rapport entre les informations disponibles par rapport aux informations émergentes. Enfin, nous avons appliqué notre caractérisation pour améliorer un design existant. Ces travaux sont les premiers pas vers une approche formelle pour améliorer le processus de design et la réutilisation des visualisations [HURTER IHM 09].

### 1. Conclusion sur le design écologique

La perception écologique a été étudiée et définie par Gibson [Gibson, 1979]. Sans pouvoir le prouver, nous pensons que le design écologique est une sous-partie des visualisations qui comporte des données émergentes. En effet, avec le design écologique, l'utilisateur perçoit des informations automatiquement avec des affordances qui font référence à l'environnement naturel de l'Homme.



Les designs écologiques sont très peu utilisés dans le domaine de l'InfoVis alors qu'ils présentent un intérêt très important au regard de la diminution de la charge cognitive pour percevoir de l'information. Par exemple, l'utilisation de la comète radar montre

comment l'utilisation de la perception écologique permet d'améliorer la perception de données temporelles.

Nous avons étudié plusieurs visualisations écologiques de données temporelles :

- Designs utilisant la disparition progressive : comète radar et MAMMI (rémanence), VideoProbe (vieillessement),
- Designs utilisant l'apparition progressive et l'accumulation : alarmes, papier buvard,
- Designs utilisant le mouvement : Passage de Niveaux de Vol.

Nous avons donné une définition simple de l'émergence basée sur l'utilisation du modèle de DataFlow. Cette définition n'est pas suffisante, il nous faut encore répondre à de nombreuses questions : chaque représentation peut potentiellement faire émerger de nombreuses informations, mais combien sont réellement utilisées ? De plus l'utilisation de représentations écologiques permet une compréhension spontanée mais a un coût car elles sont rarement aussi compactes (dans le sens du nombre d'informations affichées) que les représentations "optimisées" (par définition) telles que les TreeMaps qui ne sont certainement pas écologiques. Il y a, pour le concepteur, un compromis à faire entre la compréhension spontanée et l'efficacité des designs.

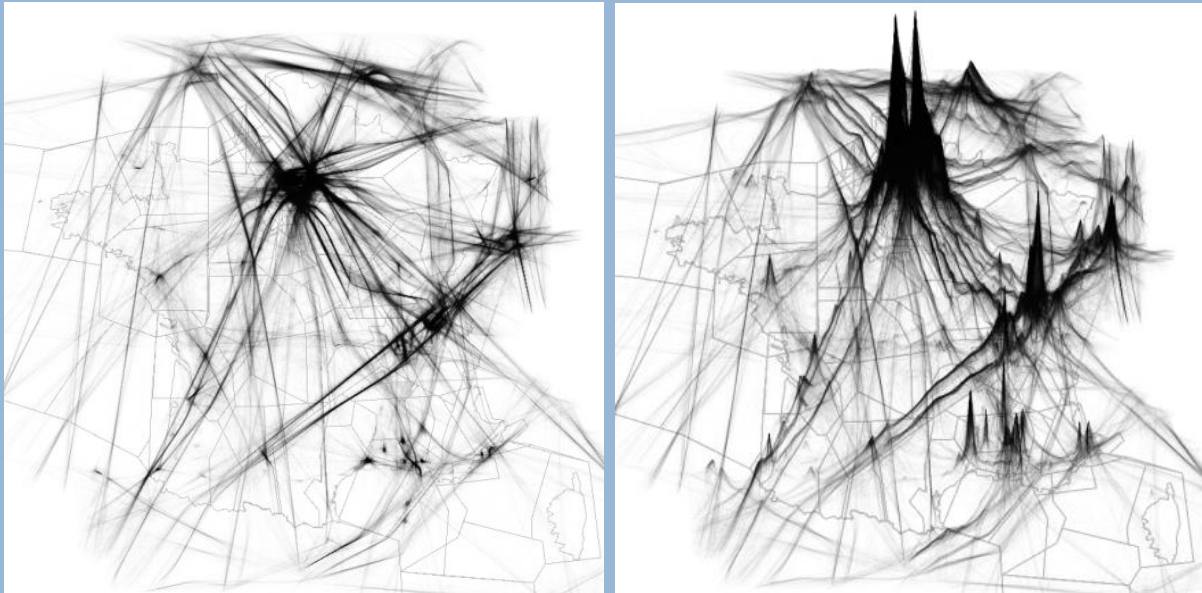
## **2. Conclusion sur la caractérisation et l'étude des designs écologiques**

Nous avons analysé plusieurs visualisations du domaine des designs écologiques. Notre démarche s'inscrit dans la réalisation d'outils pour aider les designers à la réalisation de visualisations efficaces. Nous avons affiché les différences entre les visualisations à l'aide d'un tableau pour caractériser les designs. Nous avons introduit un critère qualificatif pour mesurer l'efficacité des visualisations face au nombre de données émergentes qu'elles contiennent: le design le plus efficace est celui qui fait émerger le plus grand nombre d'informations. Ce critère ne s'applique pas qu'aux représentations écologiques, mais à toute représentation.

De plus, nous avons analysé ces représentations écologiques pour comprendre comment elles utilisent la perception écologique de l'utilisateur. Avec l'utilisation d'un modèle très simplifié de la perception, nous avons montré comment l'utilisateur perçoit l'évolution de la valeur des variables visuelles dans une représentation statique ou animée. Nous avons analysé avec des considérations géométriques les informations codées par le design du passage de niveau. Cette analyse nous a permis de valider de façon analytique la perception de l'utilisateur avec l'émergence du début et de la fin d'évolution des avions. Cette méthode peut être répliquée pour analyser d'autres designs mais elle reste assez longue.

Enfin, nous avons utilisé nos connaissances sur le design pour analyser et améliorer la visualisation du PNV en trouvant un design plus adapté grâce à l'usage de la perception écologique.

## Partie 3 : La réalisation de visualisations



Une journée de trafic aérien en utilisant la transparence (gauche) et en utilisant l'accumulation (droite). FromDaDy.

Dans cette partie, nous utilisons nos connaissances acquises dans le domaine de l'InfoVis pour produire des visualisations efficaces. Tout d'abord, nous réalisons une représentation schématique des routes aériennes. Ensuite, nous développons un logiciel pour la visualisation interactive de grandes quantités de données.



## **I. Réalisation de vues « métro » pour les routes aériennes**

---

Ce chapitre est consacré à l'application des connaissances en InfoVis et plus particulièrement à la réalisation et l'amélioration de nouvelles interfaces visuelles dans le domaine du Contrôle Aérien. Les avions doivent suivre leur plan de vol dans les espaces de Contrôle. Ce cheminement est publié par le Service d'Information Aéronautique (SIA). Les routes aériennes ainsi définies sont composées de points de référence appelés balises (spécifiées par une latitude, longitude et altitude). Aujourd'hui les plans de vols suivis par les avions ne sont pas affichés sur les écrans radar : ces routes sont trop nombreuses et leur affichage nuit à la lisibilité de la visualisation. Pourtant elles véhiculent beaucoup d'informations concernant le vol.

Partant de ce constat, nous nous sommes intéressés aux visualisations schématiques et plus particulièrement à la transposition de cartes de type métro dans le domaine du Contrôle Aérien. Les cartes de type métro sont proches de la représentation mentale utilisée par les contrôleurs aériens.

Ce sont des visualisations schématiques qui représentent la structure d'un réseau de transport avec des sections et des stations [Garland, 1994]. Depuis la première carte métro produite par Beck en 1934, les règles de design sont restées inchangées. Les lignes de métro doivent être majoritairement droites et alignées sur des angles multiples de 45° (vue octolinéaire [Nöllenburg & Wolf, 2006]). Les échelles de représentation sont dilatées dans les zones à forte densité et contractées dans les zones moins peuplées. Ainsi la distance entre chaque station est uniforme. Malgré cette distorsion géographique, la topologie générale de la vue est respectée ; les positions relatives entre les stations de métro ne doivent pas être altérées.

La génération automatique de vue métro est un domaine de recherche actif. Cependant, il n'existe aucune solution générique. La majorité des méthodes utilisées cherche à optimiser un ensemble de critères exprimés sous forme d'équations [Nöllenburg & Wolf, 2006] [Stott & Rodgers, 2004]. Les meilleurs résultats sont obtenus en utilisant des méthodes méta-heuristiques comme le recuit-simulé. Les critères utilisés sont la maximisation des directions communes, des angles octolinéaires, des lignes droites. Certaines méthodes consistent à minimiser le nombre de croisements [Bekos et al., 2007], ou encore la « simplification » des routes [Merrick & Gudmundsson, 2006].

Nous avons donc implémenté une méthode pour la production automatique de visualisations de type « métro » pour afficher de façon schématique les routes aériennes que doivent suivre les avions [HURTER IHM short 09] [HURTER AVI 10].

### **A. L'analyse en type de donnée des vues métro et carte de routes aériennes**

---

Quand un utilisateur veut prendre le métro, il cherche à répondre à cette question : « Où se trouvent les stations les plus proches ? » La réponse est une liste ordonnée (O) de stations, de la plus proche à la plus éloignée. Lorsque l'utilisateur souhaite se rendre d'un point A à un point B, il doit alors répondre à ces questions : « Quelle ligne passe par A ? Quelles lignes passent par B ? Quelle est la direction de chaque ligne ? Quelles sont les

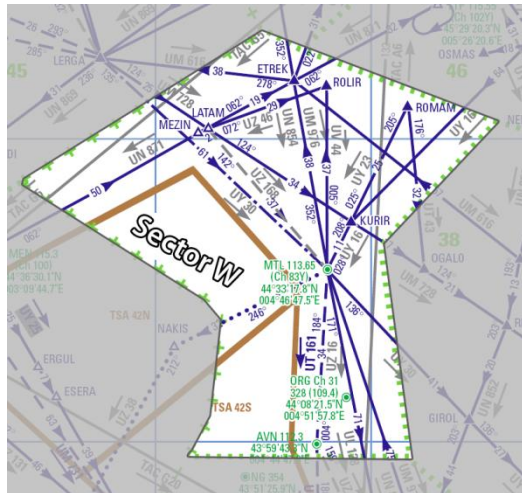
changements de lignes ? Quel est le temps estimé de trajet ? ». Les réponses à ces questions sont une liste de lignes de métro (N), leurs directions (N), les connexions (N, O) et les noms des stations (départ, changement, arrivée) (N, O pour l'estimation du temps). Quand on se déplace à l'aide du métro, les utilisateurs cherchent à connaître leur position courante et par extension savoir s'ils sont proches de leur destination. Ainsi la réponse à la question « Où suis-je ? » est le nom d'une station (N). Pour la question « Combien de stations me reste-t-il avant ma destination ? », la réponse est un nombre (Q), correspondant à la longueur d'une liste de stations ordonnées (O).



Cette analyse permet de mieux comprendre et de lister les tâches que doivent remplir les cartes de métro. De plus la décomposition en Q, O, N permet de mieux comprendre les choix de design avec la théorie des échelles [Stevens, 1946] ou la sémiologie graphique [Bertin, 1983].

### 1. Analyse de la tâche des contrôleurs aériens en la comparant aux tâches des utilisateurs des vues métro

Les contrôleurs aériens utilisent dans leur environnement des systèmes de visualisation complexes qui doivent être compréhensibles avec un minimum de charge cognitive. L'activité principale des contrôleurs est d'assurer un espacement minimum entre les avions. Les avions suivent des routes (Flight Routes : FR) publiées par les instances de régulation. Ces routes sont des séquences de balises (point géographique avec une altitude). L'image suivante montre les FRs d'un secteur de contrôle, telles que représentées sur une carte publiée.



Les routes aériennes sont très importantes pour les tâches des contrôleurs : elles permettent de prévoir les positions futures des avions, et donnent aussi une vue globale des flux de trafic aérien. Les FR sont nombreuses et s'entremêlent, c'est pourquoi leur représentation sur la vue radar est difficile : elles sont trop prégnantes par rapport à l'affichage des comètes radar. Leur représentation est incompatible avec les critères de sécurité pour les Contrôleurs Aériens. La représentation des FR à l'aide d'une vue métro est potentiellement une solution, mais ne peut pas être réalisée avec une vue métro classique, car il y a trop de différences entre les FR et les lignes physique des métros :

- Station de métro / balise aérienne : les stations sont réparties géographiquement pour maximiser le service à l'utilisateur, alors que les balises sont parfois très proches les unes des autres. De plus les réseaux formés par les FR sont bien plus hétérogènes que ceux formés par les lignes de métro (réseau en étoile ou en anneau).
- Les stations de métro sont espacées régulièrement pour assurer une desserte efficace. Les balises sont quant à elles réparties de façon non homogène le long des routes aériennes.
- Les FR ont beaucoup de croisements, car elles sont plus nombreuses que les lignes de métro.
- Les FR ont beaucoup de connexions. Les stations de métro ont rarement plus de 5 connexions.
- Les FR sont nombreuses et partagent souvent les mêmes sections mais à des altitudes différentes. En effet, La dimension verticale des routes aériennes est plus simple à exploiter que pour les lignes de métro (coûts de construction prohibitifs).

## 2. Optimisation

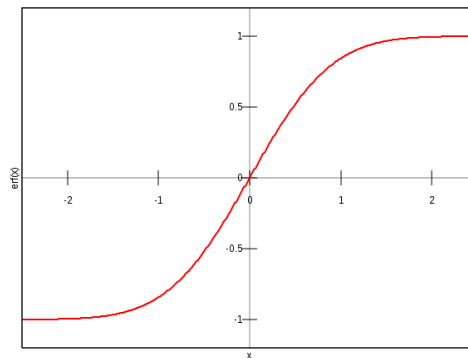
Nous avons utilisé la technique du recuit-simulé [Kirkpatrick et al., 1983]. Cette technique essaie de minimiser une liste de coûts en modifiant itérativement la visualisation. D'après l'analyse de l'activité des contrôleurs, nous avons identifié cet ensemble de coûts à minimiser :

- **Proximité topologique** : la position des balises doit être la plus proche possible de la position réelle. Si la vue produite diffère grandement de la vue radar, cela va

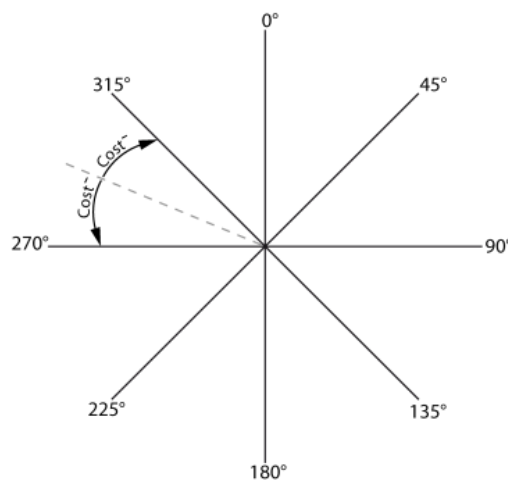
créer un décalage avec la perception mentale du contrôleur. Dans les vues métro, les stations sont souvent espacées de façon régulière, ainsi la proximité topologique est normalement respectée. Ce coût est exponentiellement croissant en fonction de la distance entre la position réelle des balises et leur représentation à l'écran.

$$Coût_{topologique} = - \frac{\sum b \in B \operatorname{erf}\left(\frac{\operatorname{Dist}(b, b') - \operatorname{Max}_{dist}}{C_{dist}}\right)}{|B|}$$

B est l'ensemble des balises du secteur, b et b' sont respectivement la position initiale et modifiée d'une balise.  $\operatorname{Dist}(b, b')$  est la distance euclidienne entre balises.  $C_{dist}$  et  $\operatorname{Max}_{dist}$  sont des constantes fixées empiriquement.  $C_{dist}$  correspond au déplacement acceptable sans pénalité et  $\operatorname{Max}_{dist}$  est la distance à partir de laquelle la pénalité est maximale. |B| est le nombre de balise du secteur. La fonction erf est la fonction d'erreur de Gauss suivante.



- **Direction** : la représentation devient plus claire quand toutes les directions des tronçons des FR sont proches de la vue octolinéaire (0, 45, 90, ..., 315, 360°). Pour chaque segment des FR, ce coût augmente linéairement en fonction de sa distance au plus proche angle octolinéaire.



$$Coût_{dir} = - \frac{\sum e \in E \frac{\operatorname{dir}(e)}{22.5}}{|E|}$$

E est l'ensemble des tronçons de route, |E| est son nombre. Dir(e) est son angle octolinéaire.

- **Courbure** : les lignes droites sont des représentations à privilégier pour afficher les routes. Le coût associé à la courbure augmente quand l'angle entre deux segments consécutifs d'une FR s'éloigne de l'angle plat (180°).

$$Coût_{courb} = \frac{\sum a \in A \frac{bend(a)}{180}}{|A|}$$

A est l'ensemble des angles entre deux tronçons de route consécutifs, |A| son cardinal. Bend(a) est la valeur de l'angle entre deux tronçons.

- **Densité** : une spécificité du domaine de l'ATC est la forte densité de balises dans certaines zones. Cette densité est incompatible avec une visualisation lisible. Considérant une balise, ce coût augmente avec la proximité d'autres balises.

$$coût_{den} = \frac{\sum b_i, b_j \in B_{i \neq j} erf\left(\frac{d(b_i, b_j) - Max_{den}}{C_{den}}\right)}{|B|^2}$$

C<sub>den</sub> et Max<sub>den</sub> sont des constantes fixées empiriquement. Max<sub>den</sub> est la distance minimale entre deux balises. La fonction d(b<sub>i</sub>, b<sub>j</sub>) est la distance entre deux balises. B est l'ensemble des balises du secteur, |B| est son nombre.

- **Croisement** : ce coût compte le nombre de croisements pour une FR (et normalisé par le nombre de tronçons). Ce critère est le même que pour la génération de vue métro classique mais avec une pondération plus forte pour le domaine ATC.

$$Coût_{crois} = - \frac{\# \text{ croisement de tronçons}}{\# \text{ tronçons}^2}$$

La visualisation suivante affiche les routes aériennes avant leur optimisation :





l'affectation des couleurs est important. Deux FR proches ne doivent pas avoir des couleurs perceptuellement proches. Ainsi les contrôleurs vont pouvoir désigner oralement une route : « la route rouge qui passe par la balise MERU » (cas de la communication entre contrôleurs), mais aussi suivre du regard une FR sur tout son trajet sans la confondre avec une autre.

La plupart des cartes de type métro est générée par des graphistes qui choisissent judicieusement le choix des couleurs et leur affectation. Pour ce faire, ils utilisent des techniques éprouvées dans le domaine du design [Ishizaki , 1995] [Rheingans, 1999] [Robertson & O'Callaghan, 1986]. Dans cette partie, nous proposons une méthode automatique pour la création d'un ensemble de couleurs discriminantes et leurs placements. Notre approche est basée sur différents travaux antérieurs [Harrower et al., 2003] [Levkowitz, 1992] [Moretti & Lyons, 2005].

## 1. GENERATION D'UNE TABLE DES COULEURS

Générer des couleurs, dans l'espace RGB en séparant les valeurs des composants, n'est pas une méthode optimale pour notre objectif : la valeur RGB ne nous renseigne pas sur la sensation de couleur perçue et la « distance » RGB de ces couleurs n'est pas homogène avec la distance perceptive. Pour définir les couleurs et mesurer efficacement leurs distances perceptives, nous utiliserons les espaces colorimétriques de la CIE (Commission Internationale de l'Eclairage) [CIE CAM 02]. La CIE a construit de manière expérimentale des modèles chromatiques théoriques, définissant l'ensemble des couleurs perceptibles, indépendamment des matériels d'affichage, et prenant en compte la perception visuelle humaine. Le modèle CIE  $L^*a^*b^*$  (CIELAB) est ainsi homogène avec la perception visuelle. La distance géométrique entre deux couleurs dans l'espace  $L^*a^*b^*$  correspond à leur distance perceptive et peut être calculée (CIE  $\Delta_e$  2000). Enfin, la variante LCHab du modèle Lab permet d'exprimer la couleur selon trois dimensions plus intuitives : Luminosité, Chroma (saturation) et Hue (teinte). Les dimensions L, C et H sont orthogonales et perceptuellement uniformes. Cependant, le modèle CIELAB a ses limites, le Gamut RGB (couleurs réellement affichables) du périphérique ne permettra pas d'atteindre toutes les couleurs LCHab possibles et le modèle ne précise pas la notion de « couleur nommée ».

En utilisant le modèle CIE LCHab, nous pouvons aisément calculer un nombre quelconque de couleurs régulièrement espacées sur l'axe « H » des teintes. Cet axe des teintes est en effet primordial pour une différenciation de sensation de couleur. Nous pouvons aussi calculer leurs contrastes relatifs ( $\Delta_e$ ), mais cette approche n'est pas suffisante car des valeurs constantes en luminosité (L) et en saturation (C) ne suffisent pas pour produire des couleurs expressives (nommables) ni garantir leur affichage dans le Gamut RGB. Comme l'ont formulé Goethe et plus précisément Munsell [Munsell, 1912], à chaque angle de teinte correspond un espace de distribution non homogène des couleurs sur les axes L et C. La luminosité du jaune ( $L > 90\%$ ) et du violet ( $L < 40\%$ ) par exemple sont très différentes. Des travaux et expérimentations sur les « couleurs nommées » [Berlin & Kay, 1969] [Párraga et al., 2008], définissent leurs coordonnées « idéales ». En utilisant ces résultats et la table de Munsell, nous avons construit une courbe de correspondance des valeurs L de ces couleurs nommées selon l'angle H des teintes.

Le pseudo code suivant présente la fonction de création d'un jeu de couleur RGB. Cette fonction prend en argument le nombre de couleurs requises, le modèle de couleur (par défaut LCHab), le profil ICC cible (par défaut sRGB), ainsi qu'un angle de teinte



optionnel, de « calage ». La fonction *namedColorLum* retourne la valeur L de luminosité « idéale » pour un angle H de teinte dans l'espace couleur spécifié. La fonction *maxChromaInRGBgamut* calcule la valeur C de saturation maximale d'une couleur d'angle H et de luminosité L, dans le Gamut RGB d'un périphérique cible (ICCprofil). Enfin la fonction de conversion *LCHcolor2RGB* retourne les valeurs RGB de la couleur LCH avec le profil ICC cible.

```
createRGBColorset (numColors, colorModel, ICCprofil, startAngle,extend)
{
  colorModel = 'LCHab' if not defined
  ICCprofil = 'sRGB' if not defined
  startAngle = 0.0 if not defined
  extend = 360.0 if not defined
  stepAngle = extend / numColors

  RGBcolors = array of numColors color
  foreach i ∈ {1 . . numColors}
  {
    newH = startAngle + stepAngle * i
    newL = namedColorLum(newH,colorModel)
    newC = maxChromaInRGBgamut(newH,newL,colorModel,ICCprofil)
    RGBcolors[i] = LCHcolor2RGB([newL,newC,newH], colorModel,ICCprofil)
  }
  return RGBcolors
}
```

## 2. Le placement des couleurs

En considérant que l'on possède une vue métro avec N lignes, la méthode décrite dans la section précédente nous permet de générer N couleurs distinctes. Cependant, il existe  $N!(N \text{ factoriel})$  choix d'associations possibles entre N couleurs et N lignes. Il convient alors de définir une mesure de la qualité d'un placement de couleur, puis de proposer une méthode capable, si possible, de trouver le placement optimal par rapport à cette mesure. Intuitivement, on cherche à associer aux lignes les plus proches les couleurs les plus distinctes du point de vue de la perception. Nous avons besoin de définir une distance entre les couleurs. Nous avons choisi d'utiliser le  $\Delta_e$ .

Pour calculer la distance entre les lignes, on définit une ligne comme une succession de points régulièrement espacés (pour ce faire, on discrétise chaque ligne).

Soit  $l_1$  une ligne, son ensemble de points est défini par :  $l_1 = \{ Pt_{11}, \dots, Pt_{1m} \}$

La distance classique entre deux lignes est la distance minimale entre les points des deux lignes. Cette distance n'est pas convenable du point de vue perceptif, car, par exemple, deux lignes qui se croisent perpendiculairement ont une distance nulle alors que perceptuellement elles sont très distinctes. Nous allons utiliser une distance point par point qui sera plus représentative du critère de discrimination humaine.

Soit  $l_1 = \{ Pt_{11}, \dots, Pt_{1m} \}$  et  $l_2 = \{ Pt_{21}, \dots, Pt_{2q} \}$  avec  $m < q$

$$dist(l_1, l_2) = \frac{(\sum_{i=1}^m \min(d(Pt_{1i}, Pt_{2j}), j = 1 \rightarrow q))}{m}$$

Où  $d$  est la distance euclidienne entre deux points. Ainsi, pour chaque point de la ligne  $l_1$ , la distance de ce point à la ligne  $l_2$  est égale à la distance minimale qui le sépare d'un point de cette ligne ; la distance entre les deux lignes est alors égale à la somme de ces distances. On normalise ensuite cette distance en la divisant par la plus grande distance entre les  $N$  lignes.

Un placement de couleur  $P$  est un ensemble de  $N$  couples de ligne et de couleur  $(l_i, C_i)$ . Nous définissons la qualité d'un placement de couleur comme suit :

$$Coût = \sum_{i,j,i < j}^N Delta_e(C_i, C_j) \times 2 \times (1 - \frac{1}{1 + e^{-dist(l_i, l_j)}})$$

Ce coût correspond à une moyenne pondérée des  $Delta_e$ . Le poids associé à chaque  $Delta_e$  dépend de la distance entre les lignes. On applique une fonction de type sigmoïde à cette distance de sorte que son poids tende vers 1 quand la distance tend vers 0 (deux lignes proches), et vers 0 quand la distance tend vers 1 (deux lignes les plus éloignées). Ainsi, l'influence du  $Delta_e$  est d'autant plus élevée que les lignes sont proches.

Trouver le placement de couleur optimal parmi  $N!$  placements possibles est un problème NP complet, donc insoluble par des méthodes déterministes. Nous utilisons la méthode du recuit simulé [Kirkpatrick et al., 1983] qui est très efficace pour ce type de problème. Cette méthode consiste, à partir d'une configuration initiale, à faire évoluer par voisinage. Le passage d'un état courant à son voisin se fait de deux manières : soit le voisin augmente la fonction d'évaluation et il alors est automatiquement accepté, soit il la fait décroître induisant une acceptation avec une probabilité qui dépend de l'amplitude de cette décroissance et d'un paramètre empirique dit « température  $T$  » qui diminue à chaque itération. Cette méthode ne permet pas forcément de trouver la solution optimale, mais permet d'éviter les minima locaux et ainsi de converger vers une solution acceptable.

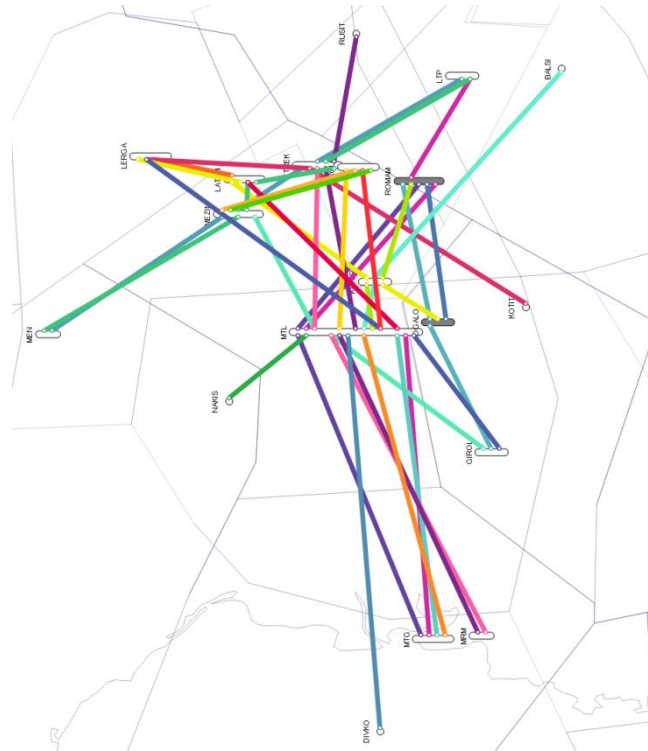
Dans notre cas, l'espace étudié est celui du placement des couleurs préalablement calculées : un état de cet espace est un ensemble de couples (route, couleur). On obtient un voisin en permutant les couleurs associées à deux routes. Afin de rendre le paramétrage du recuit simulé indépendant des FR à optimiser (pour augmenter la généralité de la méthode), nous normalisons la fonction de coût.

## C. Visualisations réalisées

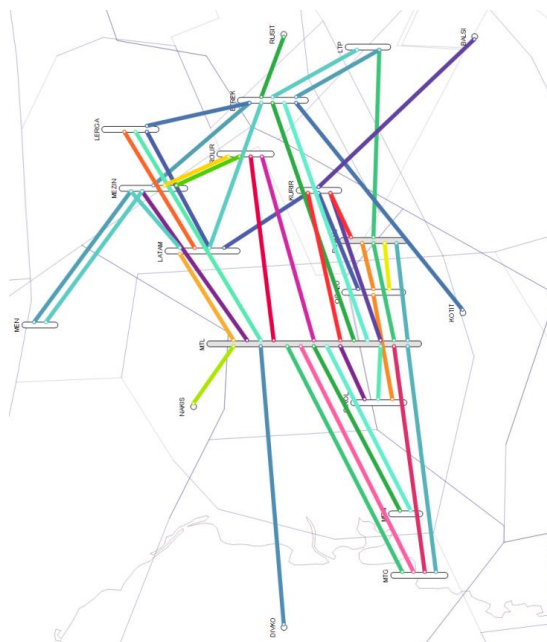
---

Nous avons appliqué notre méthode d'optimisation géographique et de placement de couleurs à plusieurs secteurs de contrôle. Ainsi pour un secteur nous avons créé quatre visualisations différentes : avec ou sans optimisation géographique (étalement des balises et décroisement), avec le placement optimal de couleur ou avec le placement le plus défavorable des couleurs. Pour calculer le placement défavorable des couleurs, il suffit d'inverser la fonction de coût pour le calcul de la distance perceptive entre deux couleurs.

La visualisation suivante affiche les routes aériennes avec leur position géographique originale (pas d'optimisation géographique), mais avec un placement de couleurs optimal (les routes proches ont des couleurs très distantes perceptuellement).

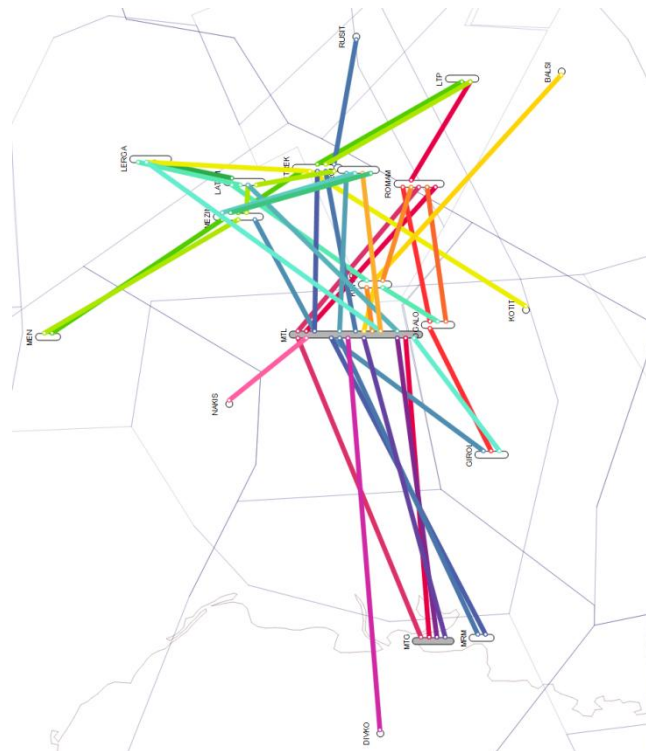


La visualisation suivante affiche les routes aériennes avec une optimisation sur leur placement (étalement des balises et décroisement des routes) et avec un placement de couleurs optimal (les routes proches ont des couleurs très distantes perceptuellement). Cette visualisation est celle qui comporte le plus d'optimisations et donc supposée être la plus efficace pour afficher la structure des routes aériennes.

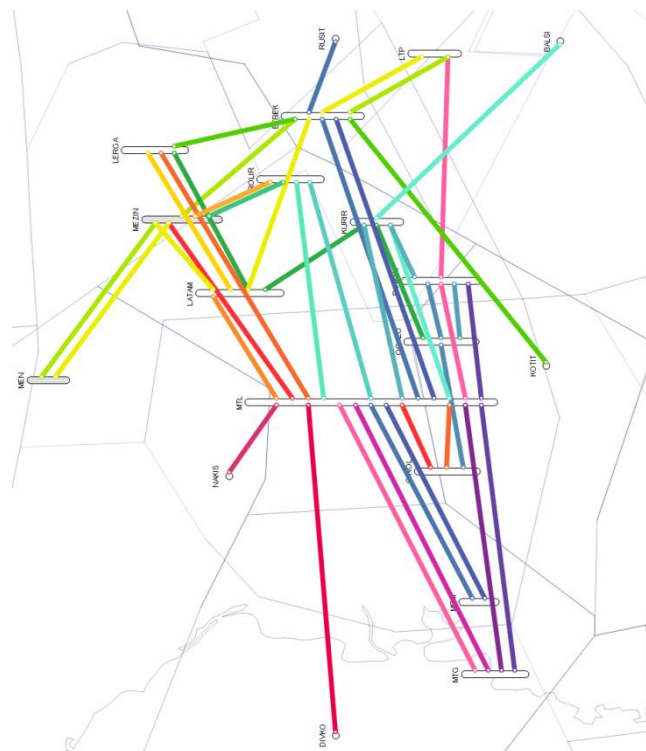


La 3ème visualisation affiche les routes aériennes avec leur position géographique originale (pas d'optimisation géographique) et avec un placement de couleurs le plus

défavorable (les routes proches ont des couleurs très proches perceptuellement). Cette visualisation est la moins efficace.



La dernière visualisation affiche les routes aériennes avec une optimisation sur leur placement (étalement des balises et décroisement des routes) et avec un placement de couleurs les plus défavorable (les routes proches ont des couleurs très proches perceptuellement).



## D. Validation des visualisations

---

Nous avons effectué des évaluations qualitatives des visualisations avec des contrôleurs aériens pour s'assurer qu'elles ne comportaient pas d'erreurs topologiques importantes et qu'elles étaient facilement compréhensibles. Ensuite, nous avons effectué des évaluations quantitatives pour valider l'efficacité des visualisations produites.

Dans ces expérimentations, nous avons utilisé un écran 19" distant de 80 cm des utilisateurs. Notre scénario simule un cas d'utilisation où une route aérienne doit être désignée le plus rapidement possible parmi toutes les routes d'un secteur de contrôle. Nous avons utilisé les quatre types de visualisations disponibles. Nous avons choisi d'utiliser le placement des couleurs le plus défavorable comme référence, plutôt qu'un placement aléatoire, qui dans un cas rare mais de probabilité non nulle, peut produire le meilleur placement. Nous avons en plus introduit un niveau de difficulté dans la désignation des routes aériennes. L'utilisateur doit désigner une section de route que nous avons qualifiée en trois niveaux :

- Niveau 1 : facile, la section est située dans une zone non dense sans croisement avec d'autres routes.
- Niveau 2 : moyen, la section est située dans une zone dense avec moins de deux croisements avec d'autres routes.
- Niveau 3 : difficile, la section est située dans une zone dense avec plus de deux croisements avec d'autres routes.

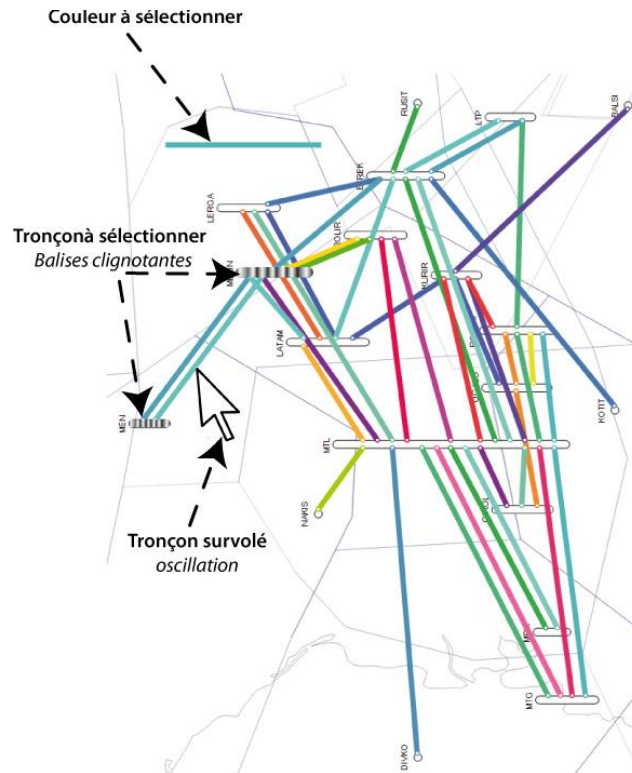
Chaque participant doit sélectionner 45 sections (5 cartes différentes, avec 9 tronçons par carte) avec deux conditions :

- le placement des couleurs *défavorables* avec ou sans optimisation géographique,
- le *meilleur placement* des couleurs avec ou sans optimisation géographique.

Au total, chaque participant (12 au total) doit sélectionner 90 tronçons dans un ordre aléatoire. Le nombre de cartes et le mélange des conditions réduisent fortement l'effet d'apprentissage. Nous avons enregistré le temps de réaction et la validité de la réponse à chaque essai.

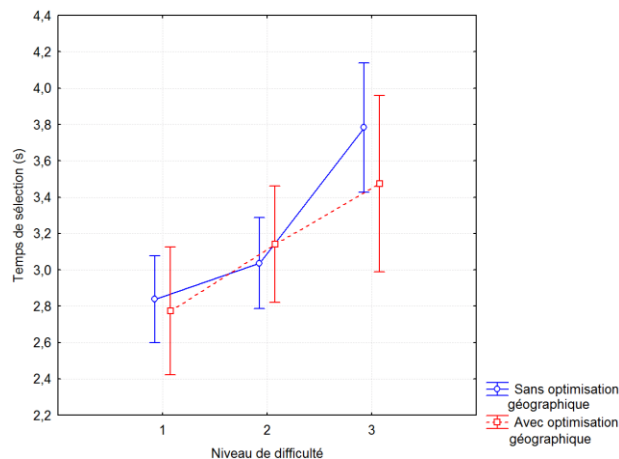
### 1. Plan expérimental

L'image suivante détaille la visualisation utilisée pour réaliser l'expérimentation. Une ligne horizontale (située dans le coin supérieur gauche de l'écran) indique la couleur de la route aérienne à sélectionner. A chaque nouvel essai, le pointeur de la souris est positionné sur cette ligne pour s'assurer que l'utilisateur mémorise sa couleur. Deux balises clignotantes indiquent la section de la route à sélectionner (utilisation de l'effet pré-attentif [Treisman, 1985]). Lorsque l'utilisateur survole une section de route aérienne, celle-ci se met à osciller et indique ainsi que l'utilisateur peut la sélectionner en appuyant sur le bouton gauche de la souris. Nous avons pensé utiliser d'autres designs mais l'oscillation est le seul choix de représentation qui ne modifie pas la perception de la couleur contrairement à la modification de la taille des routes aériennes [Tabart et al., 2007].



## 2. Résultats

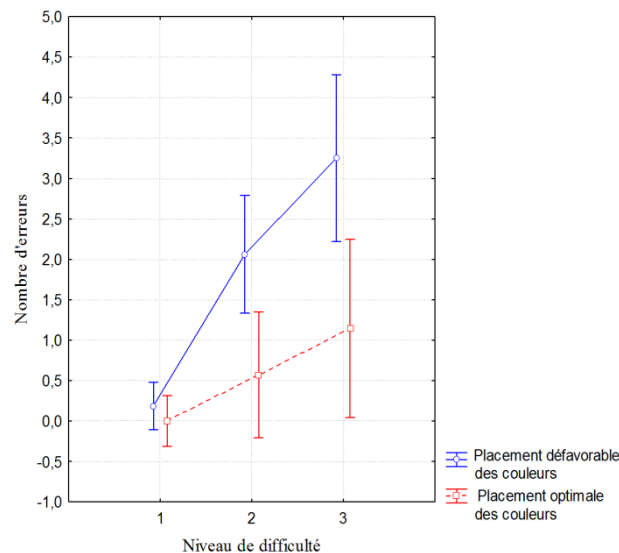
Toutes les mesures de dépendance ont été analysées à l'aide de l'analyse de variance (ANOVA) avec deux groupes de participants (avec et sans optimisation géographique) et avec des mesures répétées sur deux facteurs : le niveau de difficulté (facile, moyen, difficile) et le type d'affectation des couleurs (optimisé ou défavorable). La figure suivante donne le temps de sélection d'un tronçon en fonction de l'indice de difficulté.



Le niveau de difficulté a un impact important sur le temps de sélection d'un tronçon de route.  $F(2,28) = 45.77$   $p < .05$  ; plus l'indice de difficulté est important, plus le temps de sélection est grand. De plus, il existe une interaction entre le niveau de difficulté et la condition géographique (avec ou sans optimisation géographique),  $F(2,28) = 4.33$ ,  $p < .05$ . L'augmentation du temps de sélection due à l'indice de difficulté est moins importante avec la vue optimisée géographiquement que pour la vue non optimisée

géographiquement. Ainsi la visualisation optimisée géographiquement est plus efficace (l'utilisateur fait une sélection plus rapidement) que la visualisation non optimisée géographiquement.

Le graphique suivant montre le nombre d'erreurs en fonction de l'indice de difficulté et dans toutes les conditions de l'expérimentation.



L'analyse du nombre d'erreurs confirme l'influence de l'indice de difficulté  $F(2,28) = 8.79$   $p < .05$  : le nombre d'erreurs augmente avec la difficulté. Enfin, il existe une interaction entre le niveau de difficulté et le type de placement de couleur (optimisé ou défavorable).  $F(2,28) = 40.79$   $p < .05$ . Le nombre d'erreurs dues à l'augmentation du niveau de difficulté est moins important avec un placement de couleurs optimisées qu'avec un placement de couleurs défavorables. En ce sens, le placement des couleurs optimisé est plus efficace que le placement de couleurs défavorable.

## E. Conclusion

Dans ce chapitre, nous avons étudié la classe des visualisations schématiques métro. Leur utilisation a été transposée dans le domaine du Contrôle Aérien. Nous avons traduit les critères de cette classe de visualisations en équation mathématique, et avec la méthode du recuit simulé, nous avons produit automatiquement des visualisations. Nous avons proposé et validé une méthode pour générer et placer des couleurs sur ces visualisations. Notre placement géographique et notre affectation des couleurs permettent de produire des visualisations plus lisibles et plus efficaces pour une tâche de sélection de section de route aérienne.

Bien que nous l'ayons appliqué au cas spécifique de la représentation des routes aériennes, notre démarche peut s'appliquer au domaine des cartes de transports urbains.



## II. Interaction avec les données

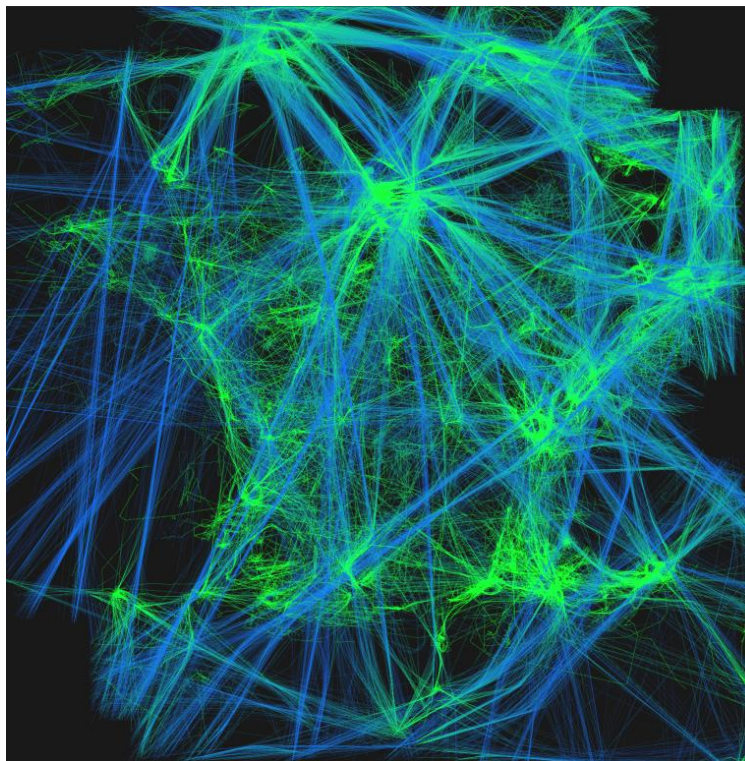
---

Dans cette partie, nous détaillons nos travaux de conception et de réalisation d'un logiciel pour l'exploration de grandes quantités de données. Ce système complète l'ensemble des logiciels d'exploration de données existants en intégrant des fonctionnalités innovantes qui permettent à la fois la visualisation et la manipulation de trajectoires.

### A. L'exploration de données dans le domaine de l'Aviation Civile

---

L'analyse des trajectoires d'avions est une activité courante dans le domaine de l'ATC, mais la densité et le nombre de trajectoires rendent cette analyse difficile. Par exemple, une journée d'enregistrement radar représente 20 000 trajectoires qui se croisent et s'entremêlent. L'image suivante a été réalisée avec les données d'une journée de trafic enregistrée au dessus de la France. Le dégradé de couleurs du vert au bleu correspond à des altitudes croissantes de trajectoires. Les côtes françaises émergent de cette visualisation grâce aux nombreuses trajectoires d'avions de tourisme le long des rivages français. Les routes aériennes en altitudes élevées émergent avec de grandes lignes droites en bleu.



Pour conserver l'interactivité lors de l'exploration des données, le système de visualisation doit avoir un temps de réponse (temps entre deux images différentes) inférieur à 100 ms [Shneiderman, 1987]. 20 000 trajectoires représentent une masse conséquente d'environ 1 million de points. En plus du problème de taille, l'utilisateur doit s'affranchir des erreurs et des incertitudes contenues dans le jeu de données. Certains enregistrements radar peuvent manquer : perte de couverture radar, données erronées



(problème de transmission bord-sol : garbling). Ces éléments doivent être pris en compte pour rendre efficace l'exploration de données.

Nous avons développé FromDaDy (From Data to DisplaY), un logiciel d'aide à l'exploration de données qui permet de résoudre les problèmes de visualisation et d'interaction avec de grandes quantités de données contenant un certain nombre de défauts. FromDaDy utilise des paradigmes très simples d'interaction pour l'exploration de données multi-dimensionnelles avec l'utilisation d'un repère cartésien, d'un outil de brush pour la sélection, de vues juxtaposables, et d'un ensemble d'outils de configuration rapides pour les choix de design. L'aspect fondamental et innovant de FromDaDy, comparativement aux outils d'exploration actuels, est la possibilité offerte à l'utilisateur *d'étaler* les données dans un ensemble de vues. En d'autres termes, l'utilisateur ne peut pas avoir deux fois la même trajectoire présentée dans son espace de travail. Ainsi, avec un ensemble d'outils de design et d'interactions efficaces, l'utilisateur est capable de filtrer, effacer, ajouter des trajectoires jusqu'à l'obtention d'un sous-ensemble de trajectoires correspondant à une requête complexe et difficile à formuler par d'autres moyens.

## **B. Cahier des charges pour l'exploration de trajectoires**

---

Pour réaliser une exploration efficace de données contenant des trajectoires avec des erreurs, l'utilisateur doit pouvoir trouver rapidement un ensemble de trajectoires qui correspondent à des critères prédéfinis. Cette approche est différente de la plupart des tâches réalisées dans le domaine de l'InfoVis où l'utilisateur recherche des corrélations entre les dimensions des données, ou encore des éléments qui se démarquent des vues (« outliers »). Les caractéristiques des trajectoires sont difficiles à définir : les spécificités des trajectoires sont souvent exprimables avec des caractéristiques visuelles (segment droit, courbure, ou forme particulière). De plus, l'utilisateur conçoit aussi sa requête en explorant les données. Par exemple, au cours de sa recherche de trajectoires, l'utilisateur s'aperçoit que son ensemble de trajectoires sélectionnées n'est pas tout à fait correct. Il doit alors modifier sa requête, c'est à dire repenser sa stratégie d'exploration. Le système doit permettre la réalisation de requêtes itératives avec des représentations visuelles des données. Le système doit aussi permettre le changement rapide des choix de design (par exemple le choix des axes utilisés). Souvent, les trajectoires qui intéressent l'utilisateur ne peuvent pas être exprimées par intention : le système doit être capable d'assurer l'itération sur le design de la vue, par exemple de sauvegarder un état d'une sélection pour pouvoir l'améliorer plus tard. Les trajectoires forment un ensemble conséquent de données multidimensionnelles. Le système doit pouvoir les afficher avec un taux de rafraîchissement acceptable par l'utilisateur. Le dernier point concerne les erreurs et les incertitudes contenues dans les données. L'utilisateur doit pouvoir les différencier, les comprendre et les éliminer si besoin.

## **C. Les données radar à explorer**

---

Les enregistrements radar sont composés de plusieurs paramètres résumés dans le tableau suivant.

Champs	Détails
Latitude	Latitude de l'avion
Longitude	Longitude de l'avion
Altitude	Altitude de l'avion
Heure	Heure de l'enregistrement
Vitesse	Vitesse de l'avion
Identifiant	Identifiant unique de l'avion

Ces données contiennent de nombreux autres champs mais, pour des raisons de clarté, nous n'avons retenu que les plus importants. Tous ces enregistrements sont liés par l'identifiant de l'avion pour former sa trajectoire. Le système de régulation aéronautique affecte un numéro à quatre chiffres pour chaque vol. C'est le code transpondeur. Ce code est transmis par le contrôleur aérien à l'avion concerné. Celui-ci l'envoie à chaque radar pour permettre son identification.

Les trajectoires contiennent des erreurs et des incertitudes :

- Le système de détection radar est moins précis lorsque les avions ont une altitude faible.
- Les systèmes embarqués peuvent envoyer des informations fausses (mauvaise identification de l'avion, altitude incorrecte...).
- La route aérienne utilisée par un avion ne correspond pas toujours à sa route initialement planifiée (conditions météorologiques défavorables, optimisation du trafic...).

Dans certains cas, les erreurs peuvent porter de l'information : elles permettent par exemple de déterminer qu'une zone a une mauvaise couverture radar, ou qu'un avion a un problème avec ses équipements de bord. Ce genre d'erreur est facilement détectable visuellement car ces erreurs créent des discontinuités dans les données et des points marginaux (outliers) : par exemple, il arrive que l'altitude d'un avion passe subitement à une valeur très éloignée puis revienne à sa valeur initiale.

Les données contiennent aussi des incertitudes. Ceci est dû à la fréquence d'échantillonnage des données. Notre base de données contient la position d'un avion toutes les 4 minutes. La position de l'avion entre ces deux instants est inconnue. Il en résulte par exemple des erreurs lorsque l'avion atterrit : sa trajectoire peut s'arrêter quatre minutes avant de toucher le sol.

## **D. Description du système**

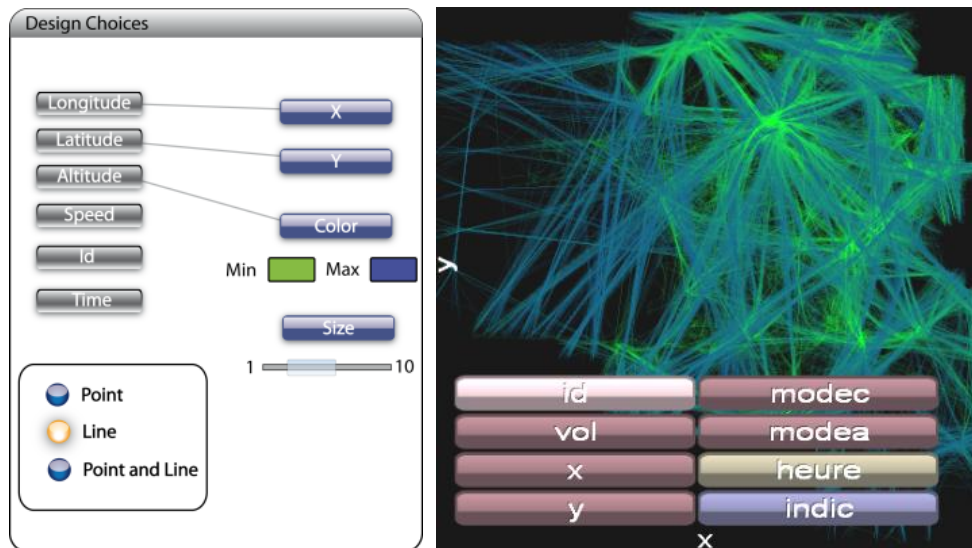
---

Cette partie détaille les fonctionnalités de base pour l'utilisation de FromDaDy. L'utilisateur peut modifier le design de sa vue, organiser les vues, faire des transitions progressives entre configurations visuelles, sélectionner et manipuler les données entre les vues.

### **1. Configuration visuelle**

FromDaDy utilise le modèle de Data Flow [Card et al., 1999] au travers d'un outil qui permet à l'utilisateur de dessiner des connexions entre les champs de données et les

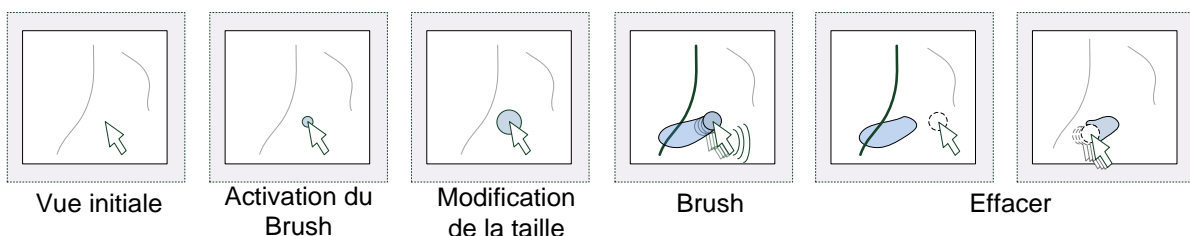
variables visuelles [Bertin, 1983] et ainsi créer des *configurations visuelles*. Par exemple, dans l'image de gauche de la figure suivante, l'utilisateur a connecté la longitude à l'axe X et la latitude à l'axe Y de la vue. L'utilisateur a aussi connecté l'altitude de l'avion avec la couleur. Ainsi les lignes de couleur bleue correspondent aux avions qui ont volé à de hautes altitudes et les lignes vertes aux avions qui ont volé dans les basses couches. L'utilisateur peut aussi choisir de double-cliquer sur un axe pour faire apparaître le menu des champs de données et ainsi choisir le nouveau champ à afficher.



FromDady utilise une mise à l'échelle automatique pour que toutes les données soient visibles à l'écran. Ce mécanisme est basé sur un calcul des valeurs minimales et maximales de tous les champs du Dataset mais aussi sur les données présentes dans chaque vue.

## 2. Le brushing et la sélection incrémentale

L'utilisateur peut sélectionner une sous-partie du Dataset en utilisant la technique de brush. Le brush est une interaction qui permet de sélectionner des entités visuelles en les peignant avec une brosse [Becker & Cleveland, 1987] dont la taille est configurable. Chaque trajectoire qui touche la zone ainsi peinte est alors sélectionnée et change de couleur. Les trajectoires sélectionnées deviennent grises et ainsi moins prégnantes dans la visualisation.

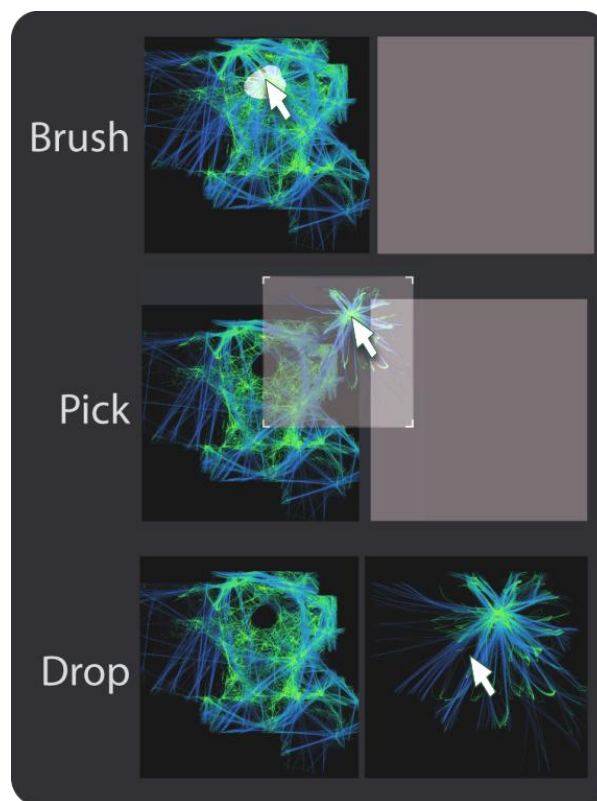


La zone de brush peut être modifiée en ajoutant de nouvelles zones (touche « Ctrl » maintenue), ou en supprimant des zones (touche « Shift » maintenue). Notre implémentation laisse une trace du brush à l'écran, permettant à l'utilisateur de visualiser et de se rappeler plus facilement la sélection qu'il a opérée. Toutes les trajectoires qui touchent la zone de brush sont sélectionnées. Ainsi, modifier la sélection revient à peindre ou à effacer cette trace. Tout en maintenant la touche Ctrl ou Shift

enfoncée, la taille de l'outil de brush peut être ajustée à l'aide de la molette de la souris. Si aucune touche n'est pressée, la molette de la souris sert à zoomer ou à dé-zoomer dans la vue. La combinaison des événements et la facilité de la modification des zones brushées, le choix de la taille de brush et le zoom de la vue permettent une action utilisateur très rapide et très efficace dans le processus de sélection.

### 3. Le paradigme « Pick and Drop »

Grâce à la technique de brushing, l'utilisateur peut sélectionner une partie des données. En appuyant sur la barre espace, il peut extraire les données précédemment sélectionnées : par défaut, les données sélectionnées sont supprimées de la vue d'origine lors du « Pick » et elles apparaissent attachées au curseur de la souris dans une vue avec un fond gris transparent. Quand l'utilisateur appuie de nouveau sur la touche espace, l'événement « Drop » s'opère. Si le curseur de la souris est au-dessus d'un espace vide, les données sont alors déplacées dans la nouvelle vue. Sinon, les données sont alors ajoutées à la vue survolée. Cette opération ressemble au traditionnel « Drag and Drop », mais nous préférons utiliser le terme « Pick and Drop » [Rekimoto, 1997] car il renforce la notion de déplacement d'objet par un média (ici le pointeur de la souris et la vue attachée).



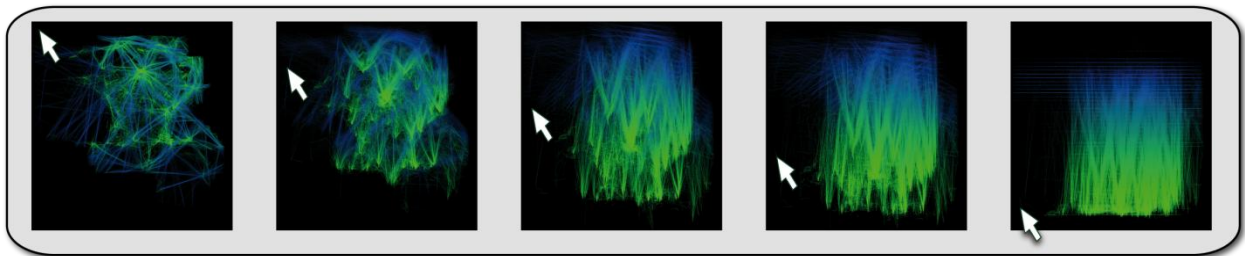
### 4. L'organisation des vues

Une session d'exploration commence avec une seule vue qui contient toutes les données, avec les axes et la configuration visuelle par défaut (configurable à l'aide d'un fichier). La vue est affichée à l'intérieur d'une fenêtre où elle occupe une cellule d'une grille infinie. Avec le « brushing » et le paradigme « Pick and Drop », l'utilisateur peut créer de nouvelles vues juxtaposées, puis changer leur configuration visuelle. L'utilisateur peut sélectionner une autre cellule pour afficher une autre sous-partie des données avec une

autre configuration visuelle. L'utilisateur peut aussi détruire une vue s'il sélectionne l'intégralité des données de cette vue et les « Pick » (en appuyant sur la barre espace) : la vue est alors vidée de toute donnée, et donc automatiquement détruite.

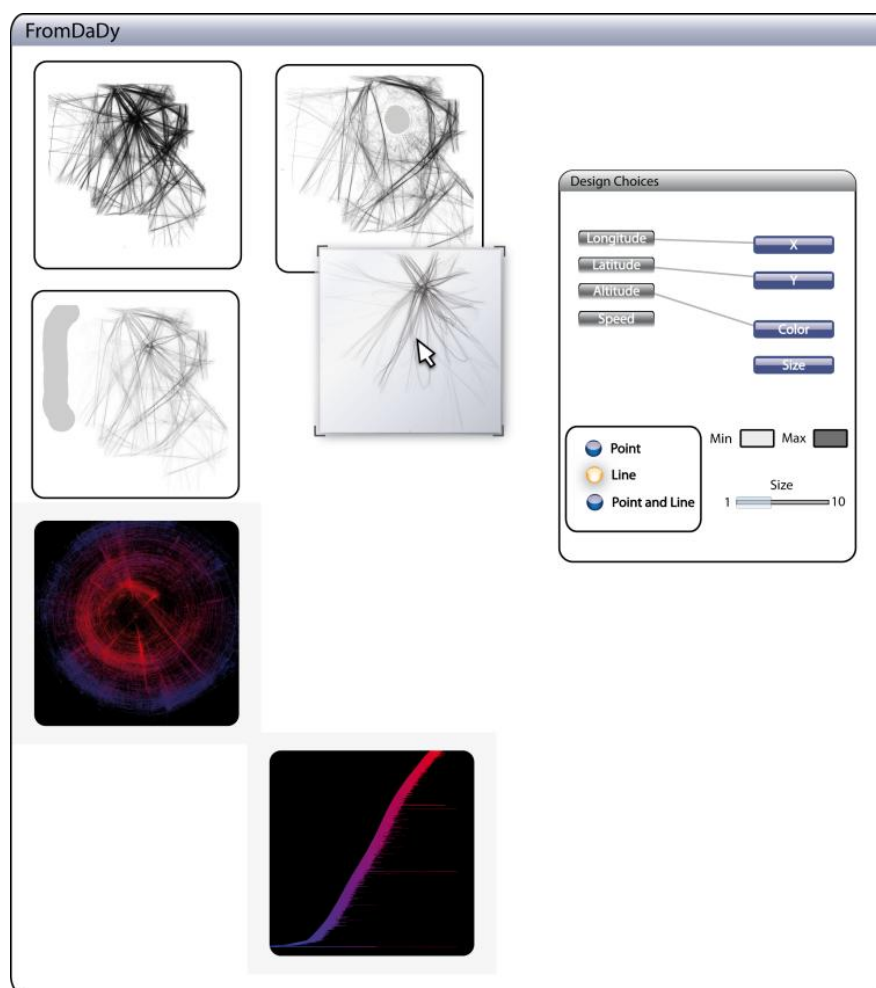
## 5. L'animation « Rolling Dice »

Les changements brutaux des axes d'une vue sont perturbants car ils empêchent l'utilisateur de garder son attention sur une entité graphique. C'est pourquoi FromDaDy utilise une animation similaire à celle de "Rolling the dice" [Elmqvist et al., 2008]. En d'autres termes, une dimension de la vue active est conservée lorsque l'autre change. Au lieu de faire une simple interpolation linéaire entre la position de chaque point dans le repère initial et le repère final, FromDaDy va faire une rotation autour de l'axe correspondant à la dimension à conserver, en utilisant une troisième dimension dont l'axe est associé à la nouvelle dimension à visualiser. Cette animation donne de la sémantique au mouvement des points, ce qui permet à l'utilisateur de l'interpréter comme une vue en rotation, et ainsi pouvoir garder le focus sur une ou plusieurs entités visuelles pendant l'animation. L'utilisateur peut aussi contrôler la transition en cliquant avec la souris sur l'axe concerné et en maintenant le bouton enfoncé tout en glissant le long de l'axe.



Cette animation est aussi utilisée lorsque l'utilisateur survole une autre vue avec des données attachées au curseur de la souris (après un « Pick »). La vue survolée va transmettre sa configuration visuelle à la vue attachée à la souris et dans ce cas l'animation va être lancée sur les trajectoires associées à la souris.

L'image suivante permet de mieux comprendre l'agencement des fenêtres de FromDaDy avec les différentes vues organisées par les choix de l'utilisateur et la fenêtre de design qui permet de les configurer.



## E. Premier bénéfice de l'utilisation de FromDaDy

FromDaDy a été utilisé par des experts ATC dont des contrôleurs aériens. Pendant cette évaluation qualitative, nous avons observé comment ils utilisent les capacités du logiciel, et plus particulièrement l'étalement des données dans les cellules de la grille, les fonctionnalités apportées par le paradigme de « Pick and Drop », les choix offerts par la fenêtre de design et les opérations booléennes implicites.

### 1. L'étalement de données dans les cellules de la grille

Dans FromDaDy, il y a toujours une seule représentation de chaque trajectoire : les trajectoires ne sont pas dupliquées mais réparties dans les différentes cellules de la grille. Cette technique a au moins deux avantages : premièrement les données ne sont jamais détruites mais déplacées (par exemple l'utilisateur peut décider de créer une vue poubelle où il va déposer toutes les trajectoires qui ne l'intéressent pas) ; deuxièmement, cela permet de créer des sous-ensembles de données de façon incrémentale. L'utilisateur peut rapidement évaluer si les données déplacées sont intéressantes. Dans certains cas, les trajectoires déplacées découvrent d'autres données initialement cachées. Ce processus d'aller-retour assure le support incrémental de l'exploration de données.

## 2. Prévoir le résultat du Drop, et le « Design Picker »

L'animation « Rolling Dice » est aussi utilisée quand l'utilisateur déplace des trajectoires précédemment *Pickées*. Lorsqu'il survole une vue existante, la configuration visuelle de celle-ci n'est pas obligatoirement la même que celle des éléments *Pickés*. Pour éviter les changements brutaux, une rotation progressive est lancée sur la vue attachée à la souris pour que les axes correspondent à ceux de la vue survolée. De plus les couleurs se copient, ainsi que la taille, le pan et le zoom. L'animation est progressive pour faciliter la compréhension de l'utilisateur.

Cette animation permet d'ajouter deux fonctionnalités à FromDaDy. Après survol, les deux vues ont la même configuration. L'utilisateur est capable d'évaluer le résultat du Drop sans l'effectuer vraiment et donc de décider si le Drop est utile ou non. La deuxième fonction est le « Design Picker ». La vue survolée a la même configuration visuelle que la vue attachée au curseur de la souris. L'utilisateur peut alors déposer les trajectoires dans un emplacement vide, ce qui correspond à appliquer la configuration visuelle d'une vue à une autre.

## 3. L'association des trajectoires et du Brush pour faire une sélection efficace

Les trajectoires sont affichées à l'aide de points connectés entre eux par une ligne. D'autres choix de design ont été envisagés, comme l'utilisation d'une couleur, d'une forme, ou d'une taille par trajectoire. Les trajectoires sont trop nombreuses pour envisager ces choix de design, c'est pourquoi le choix d'utiliser des lignes reste le seul design convenable. Comme précédemment expliqué, le brushing sélectionne des trajectoires et non des points. Ce choix d'interaction a un avantage : dans des zones très denses (avec beaucoup de segments de trajectoires) il est très difficile de brusher un élément spécifique. FromDaDy offre deux possibilités : soit l'utilisateur itère en brushant largement puis en déplaçant le sous-ensemble de trajectoires dans une autre vue en recommençant à brusher jusqu'à isoler la trajectoire souhaitée, soit il brushe la trajectoire souhaitée dans une zone moins dense. Le zoom n'est pas forcément une solution pour la sélection dans les zones denses, car souvent l'utilisateur a besoin d'avoir une vue d'ensemble de la trajectoire concernée.

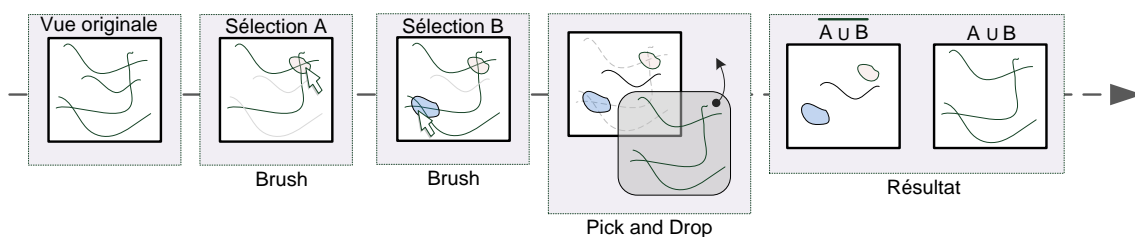
Il faut aussi savoir que le choix d'afficher des points connectés pour des lignes peut entraîner des erreurs d'interprétation des données. Par exemple, lors d'une perte de détection radar, FromDaDy va tout de même connecter par une ligne les positions antérieures et postérieures au trou de poursuite empêchant ainsi l'utilisateur de se rendre compte de la perte d'information. Ce genre de défaut est éventuellement visible avec l'apparition de segments de trajectoires linéaires trop longs.

L'exploration de trajectoires impose l'utilisation de sélections aux contours complexes, c'est pourquoi dans la majorité des cas, une sélection par rectangle n'est pas envisageable. Contrairement à d'autres logiciels de visualisation, FromDaDy utilise un unique outil de sélection à base de brush : l'utilisateur peut ajouter des zones de sélection et les supprimer. Il n'y a pas de mode d'effacement des données car le paradigme « Pick and Drop » peut suffire à déplacer des données dans une vue dite « poubelle ». Ainsi le nombre d'outils disponibles reste réduit ce qui simplifie la prise en main du logiciel.

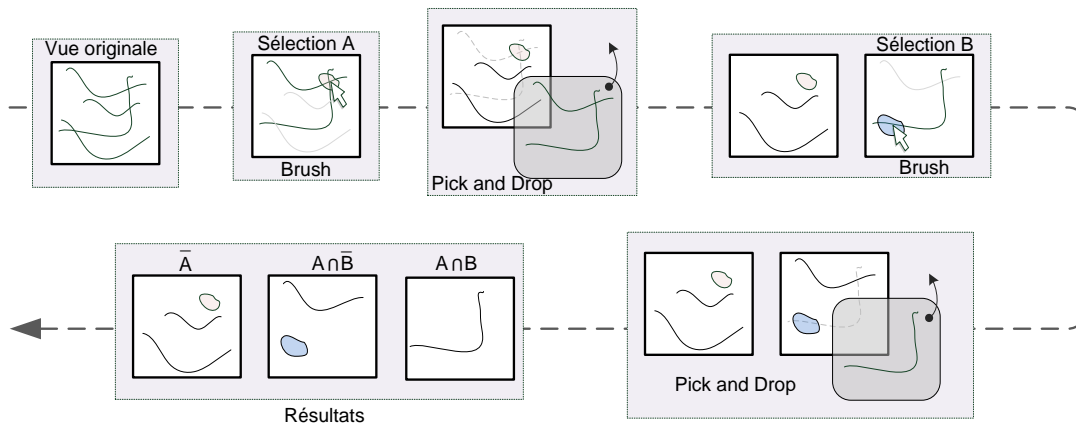


#### 4. Opérations booléennes implicites

Les opérations booléennes sont toujours complexes à réaliser pour l'utilisateur. Même avec des interfaces astucieuses, leur résultat est souvent difficile à prévoir [Young & Shneiderman, 1993]. FromDaDy réduit ce problème grâce à son paradigme d'interaction « Pick and Drop » qui crée implicitement ces opérations booléennes. Les deux exemples suivants illustrent l'union (le « Et » logique), l'intersection (le « OU » logique) et la négation (le « NON » logique). Ces opérations forment la base pour opérer toutes les combinaisons logiques comme par exemple le XOR. Dans le premier exemple, l'utilisateur veut sélectionner les trajectoires qui passent par la région A ou B. Il commence donc à brusher les régions A puis B puis opère un « Pick and Drop » dans la vue adjacente. La vue résultante correspond donc à l'ensemble des trajectoires qui passent par A ou B. La vue initiale correspond quant à elle à la négation de la requête.



Dans la figure suivante, la même procédure est appliquée pour trouver les trajectoires qui passent par la zone A et par la zone B. Pour ce faire, l'utilisateur doit séquencer deux « Pick and Drop ».



#### 5. Organisation de son espace de travail

FromDaDy donne à l'utilisateur le contrôle partiel pour l'organisation de son espace de travail. Il n'y a pas de fenêtre à créer et à manipuler, et il n'y a qu'un seul style de canevas : la grille infinie avec la juxtaposition de vues. Ceci permet des allers-retours rapides entre les vues avec des « Pick and Drop » successifs, particulièrement utiles lorsque l'utilisateur brush les vues trop rapidement ou de façon imprécise. L'outil de configuration du visuel est toujours disponible et permet de faire évoluer rapidement le design des vues. Ainsi l'utilisateur n'a pas à interrompre son processus d'exploration pour s'occuper de tâches secondaires.



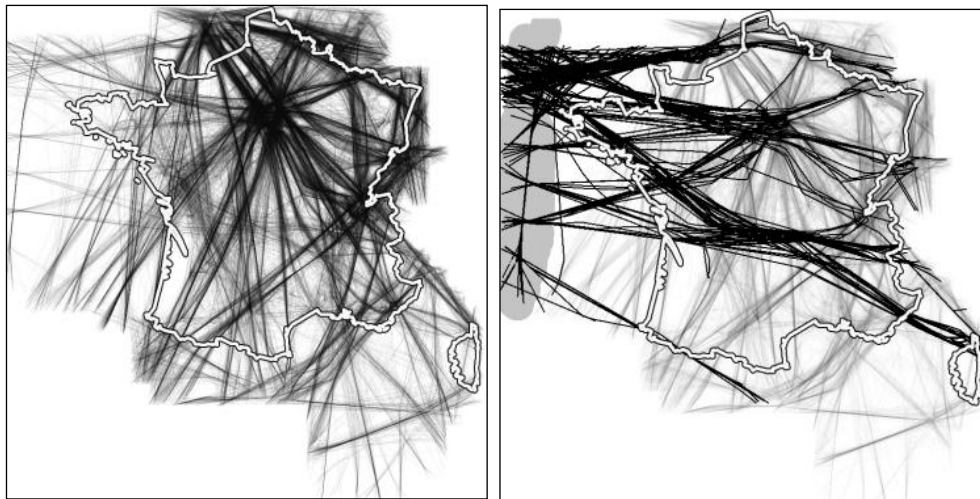
De plus, dans sa recherche de formulation de requête, l'utilisateur peut choisir d'organiser son espace de travail pour faire apparaître les différentes étapes du processus. Il en résulte un story-board. L'utilisateur peut comprendre la suite des étapes qui l'ont conduit à ne pas atteindre son objectif, et ainsi corriger sa stratégie. Ceci est particulièrement utile au cours d'une exploration infructueuse.

## F. Scénario

Cette partie présente deux scénarios qui mettent en avant les atouts de FromDaDy. Le premier scénario illustre comment explorer un ensemble de trajectoires et comment produire itérativement une requête correcte. Le deuxième scénario est un cas d'utilisation concret dans lequel FromDaDy a été utilisé pour l'apprentissage des trajectoires pour l'entraînement des contrôleurs aériens.

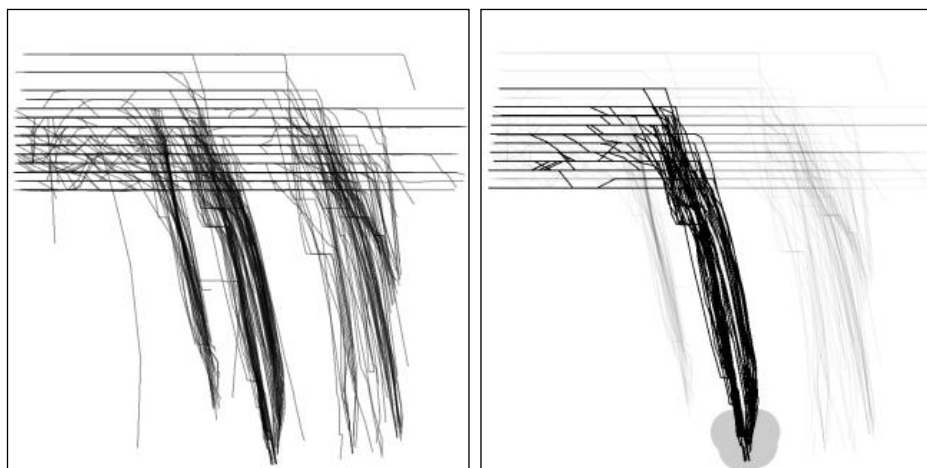
### 1. Exploration itérative

L'image suivante montre une journée d'enregistrement de vols au dessus de la France. L'utilisateur veut afficher les avions transatlantiques qui ont atterri à Roissy Charles de Gaulle. Pour répondre à cette requête, l'utilisateur établit une stratégie en deux étapes. D'abord, il filtre les avions transatlantiques en brushant la façade atlantique et ainsi sélectionne tous les avions qui ont survolé l'Atlantique ce jour.



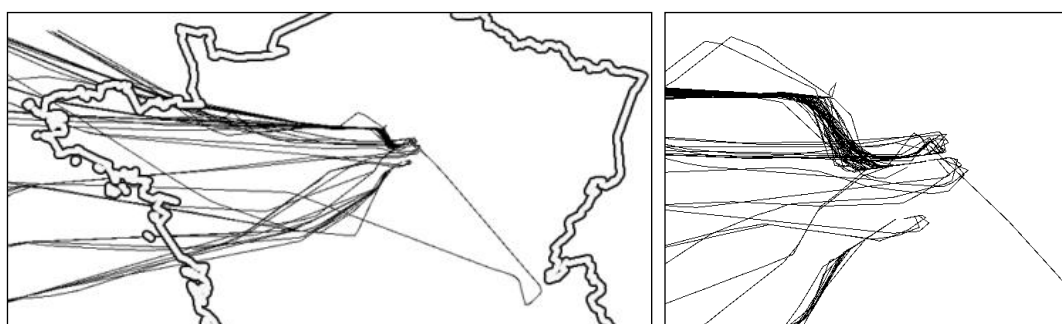
La seconde étape consiste à changer la configuration de la vue pour passer en vue verticale (Altitude, Longitude) et sélectionner les avions qui ont volé à basse altitude à la longitude de Roissy.

L'utilisateur repasse alors en vue de dessus (Latitude, Longitude) et isole ces vols à l'aide d'un « Pick and Drop ».



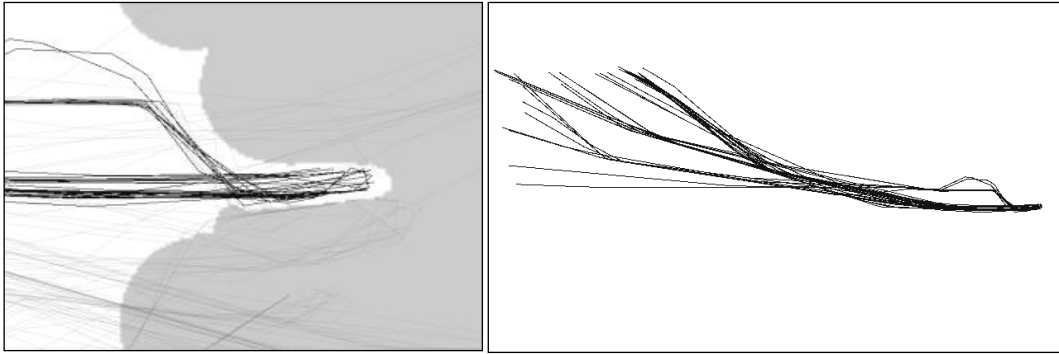
C'est alors que l'utilisateur s'aperçoit que des trajectoires isolées ne correspondent pas à sa requête initiale. En fait, certains avions sélectionnés ont atterri à Orly et non à Roissy. Ceci est dû au fait que les 2 aéroports ont la même longitude. L'étape avec la vue verticale n'est pas optimale pour sélectionner les avions atterrissant à Roissy.

Par ailleurs, l'utilisateur repère visuellement une trajectoire marginale. Après de plus amples investigations, il s'avère que cet avion a bien atterri à Roissy, mais qu'il est passé par Lyon au préalable.



Le résultat de la requête est inefficace sur plusieurs points : la sélection rate les avions dont les trajectoires qui n'arrivent pas jusqu'au sol (les données contiennent des erreurs dues au taux d'échantillonnage des trajectoires). De plus, la vue verticale oblige l'utilisateur à sélectionner toutes les trajectoires qui ont une basse altitude avec la même longitude que l'aéroport de Roissy et donc à sélectionner aussi les vols d'Orly.

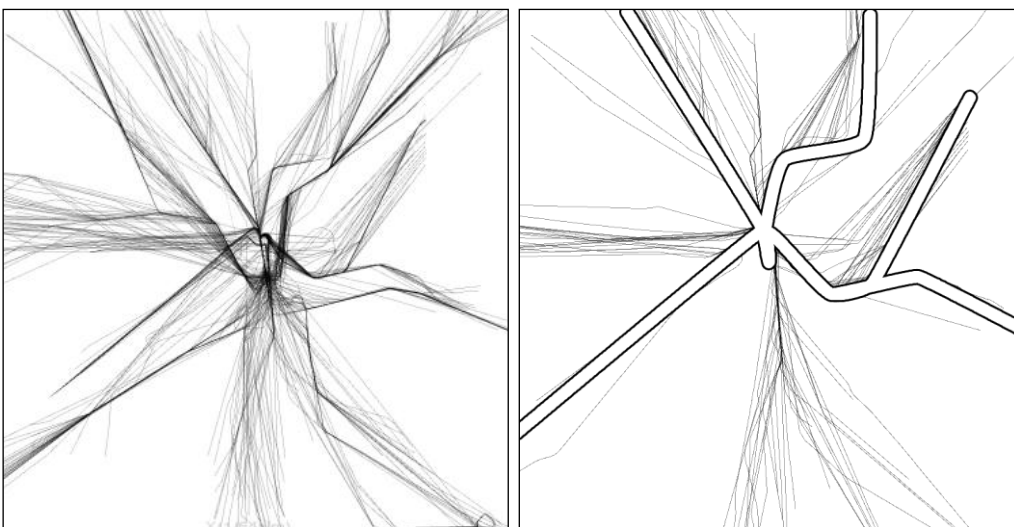
Pour toutes ces raisons, l'utilisateur doit reconsidérer sa stratégie et reformuler sa requête. Après plusieurs tentatives, il trouve un autre point de vue pour arriver à ses fins : « Les avions qui atterrissent à Roissy ne peuvent pas dépasser cet aéroport ! ». Aussi l'utilisateur sélectionne de nouveau les avions qui ont survolé l'Atlantique, puis zoom sur l'aéroport de Roissy et sélectionne tous les vols qui ont dépassé cet aéroport pour les supprimer de cette vue.



Cet exemple illustre comment l'utilisateur explore les différentes possibilités pour formuler sa requête et comment il arrive à la modifier pour finalement extraire les trajectoires qui l'intéressent.

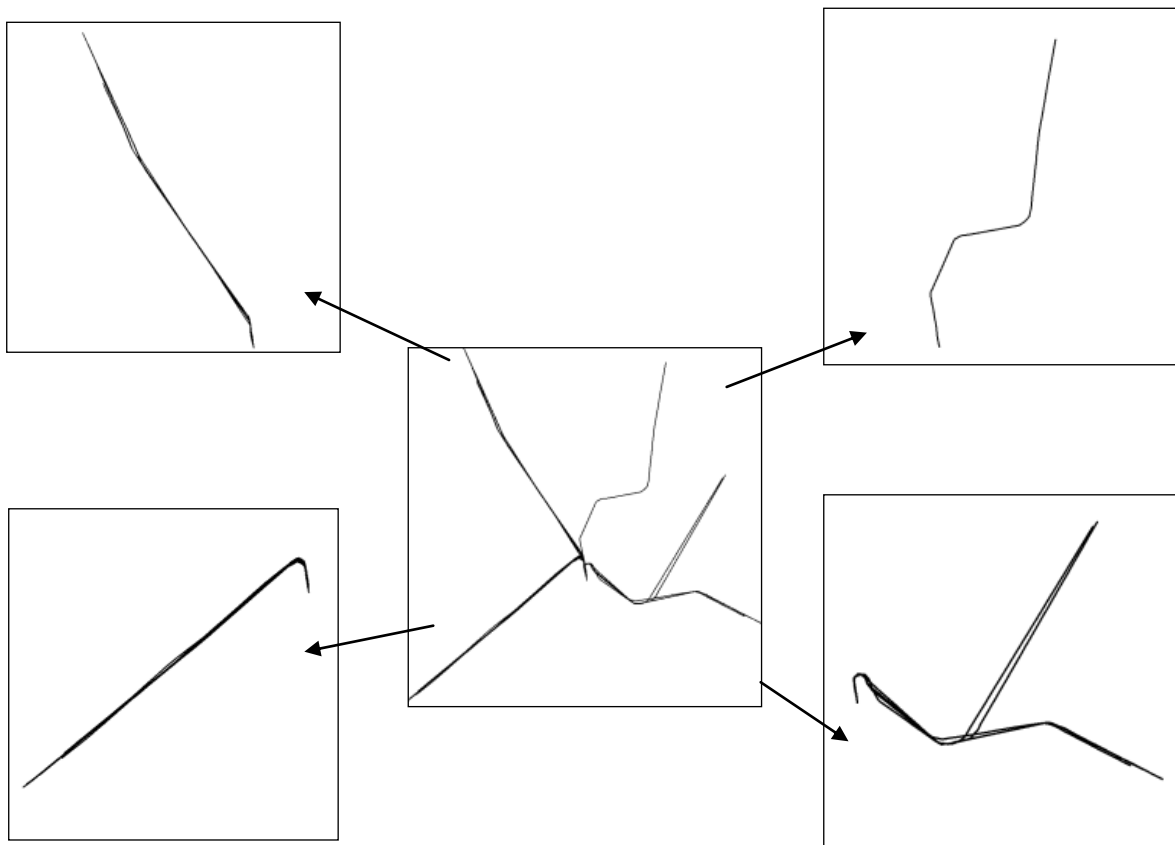
## 2. Extraction de trajectoires spécifiques pour l'entraînement des contrôleurs aériens

Cette section détaille, avec scénario réel, comment FromDaDy a été utilisé dans le cadre d'une exploration de plusieurs jours d'enregistrement de trafic aérien. L'utilisateur est un spécialiste dans le domaine du Contrôle Aérien. Sa tâche était d'extraire des trajectoires remplissant un certain nombre de critères : les avions doivent survoler des balises prédéfinies (ce qui correspond à suivre une route aérienne précise), ils doivent avoir un profil vertical particulier avec un taux de descente constant. Enfin les trajectoires d'avions à extraire doivent être triées par leur direction de départ (Nord, Sud, Est, Ouest). Ces trajectoires vont ensuite servir à d'autres systèmes pour simuler du trafic lors de séances d'entraînement des contrôleurs aériens. Toute la difficulté de cette tâche d'extraction provient du comportement non déterministe des avions. Ainsi, il est rare que les avions suivent exactement les procédures publiées. Les contrôleurs aériens altèrent les routes des avions pour des raisons de sécurité ou d'optimisation. De plus les avions dévient souvent de leur route prévue en dépassant les balises ou en anticipant un virage. L'utilisateur doit donc parvenir à extraire les trajectoires intéressantes avec des critères de sélection flous comme le profil vertical du vol. L'image suivante montre, à gauche, l'enregistrement des vols, et à droite les routes aériennes à extraire.

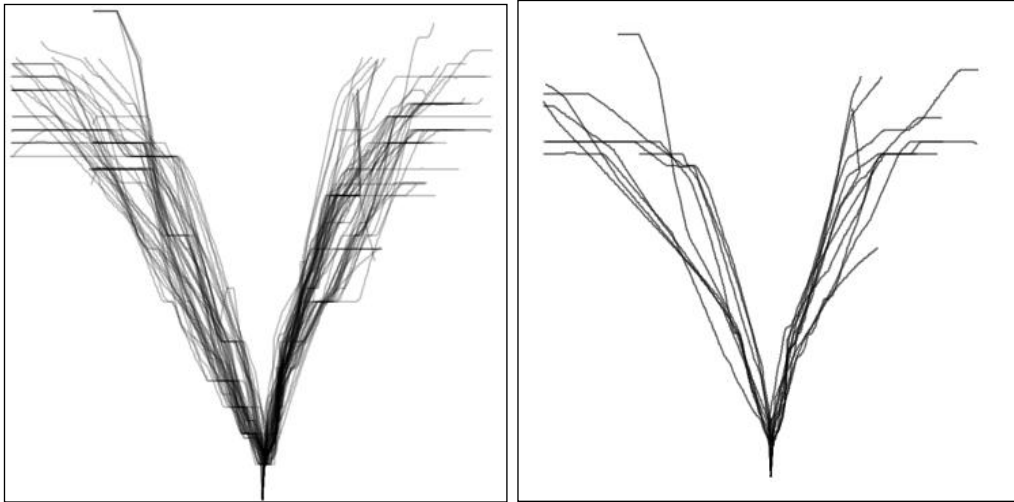


### 3. La démarche décrite pas à pas

L'image de départ est une vue de dessus du trafic d'une journée au dessus de Lyon. Chaque ligne correspond à un avion unique. L'utilisateur n'a pas besoin de connaître parfaitement les procédures standards que doivent suivre les avions car ces dernières sont très facilement détectables : elles forment des lignes plus foncées car beaucoup d'avions passent au même endroit et la transparence des trajectoires s'y accumule visuellement. Les trajectoires qui s'écartent de ces lignes noires sont souvent des avions déviés de leur route standard par les contrôleurs aériens. L'utilisateur peut rapidement les éliminer avec un *Brush*, et un « *Pick and Drop* » adapté. *FromDaDy* affiche en permanence les valeurs correspondant à la position en X et Y du curseur de la souris dans la vue, ce qui permet à l'utilisateur de *brusher* les positions exactes des balises des routes aériennes. A ce niveau, l'utilisateur crée autant de vues que de procédures identifiées (deux au nord, une à l'est et une au sud). Il *Pick et Drop* les trajectoires correspondantes dans chacune de leur vue.



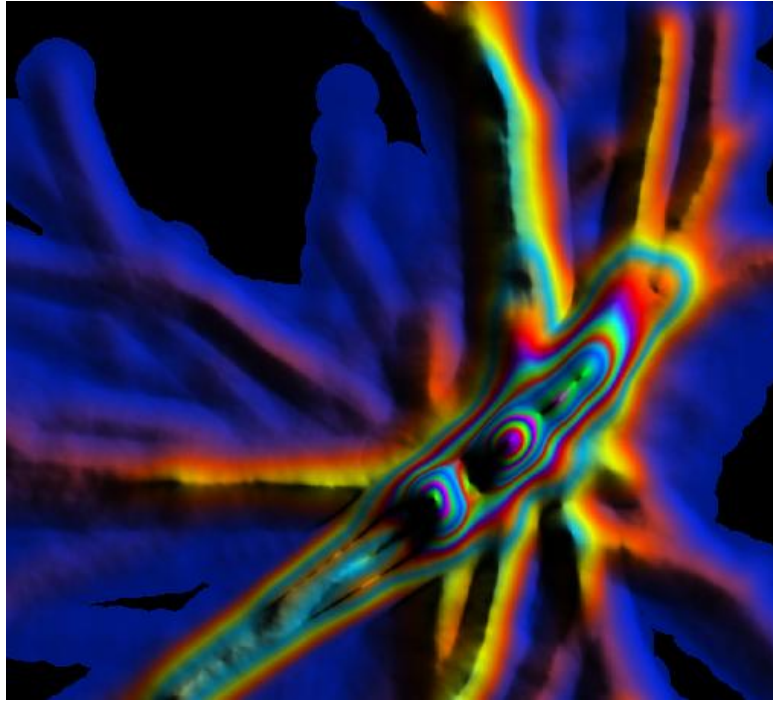
La dernière phase consiste à conserver les avions qui ont un profil vertical sans palier (vitesse verticale nulle pendant l'approche finale). Pour ce faire, l'utilisateur change sa vue pour une vue verticale (altitude, Longitude). Les avions qui ont un profil de vol avec paliers apparaissent avec une section de trajectoire horizontale. Il peut alors facilement les éliminer. Pour être complet, il change encore la configuration de vue en allant vers une représentation en (Altitude, Latitude) au lieu de (Altitude, Longitude). Ceci permet de supprimer quelques avions supplémentaires qui auraient fait des paliers avec une direction Nord-Sud et donc invisibles sur la vue précédente.



Pendant le filtrage du profil vertical, l'utilisateur s'aperçoit que l'animation « Scatterdice » peut l'aider avec une disposition des vues particulière : il copie la vue verticale (Longitude, Altitude) sous la vue de dessus (Longitude, Latitude). De cette façon, la Longitude est commune aux deux vues. L'utilisateur organise son espace de travail pour trouver la Longitude de la dernière balise de la route aérienne et ainsi à partir de quant la trajectoire doit avoir un profil vertical sans palier.

## **G. Utilisation étendue avec les cartes d'accumulation**

Lors de la transformation des données en représentations graphiques à l'aide de marques, les visualisations traditionnelles dessinent indépendamment toutes les entités visuelles. Ainsi, si deux marques ont la même position, le système remplace les pixels résultants (ou au mieux mélange les couleurs dans le cas d'utilisation de la transparence par exemple) et ainsi opère une destruction d'information. Les cartes d'accumulation (ou leur généralisation Kernel Density Estimation, KDE maps) sont des extensions du processus de visualisation. Elles permettent de mémoriser le nombre de marques qui ont rempli chaque pixel de l'écran. A titre d'exemple, les cartes d'altitude (Height maps) utilisent le processus d'illumination pour afficher les valeurs d'accumulation avec des « montagnes » et des « vallées ».



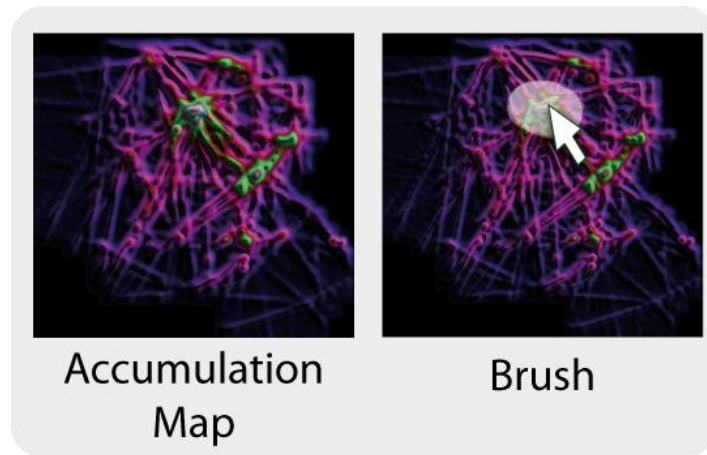
Ces cartes d'altitude utilisent une configuration implicite : la valeur de chaque pixel de l'image résultante est calculée avec une somme des valeurs d'accumulation du voisinage et est utilisée avec un éclairage pour calculer la luminosité de chaque pixel.

Dans cette partie, nous proposons un système capable de rendre explicite l'utilisation de l'accumulation et ainsi d'augmenter les possibilités de design offertes au concepteur. De plus, nous avons étendu les possibilités de FromDaDy pour permettre l'interaction avec ces cartes d'accumulation :

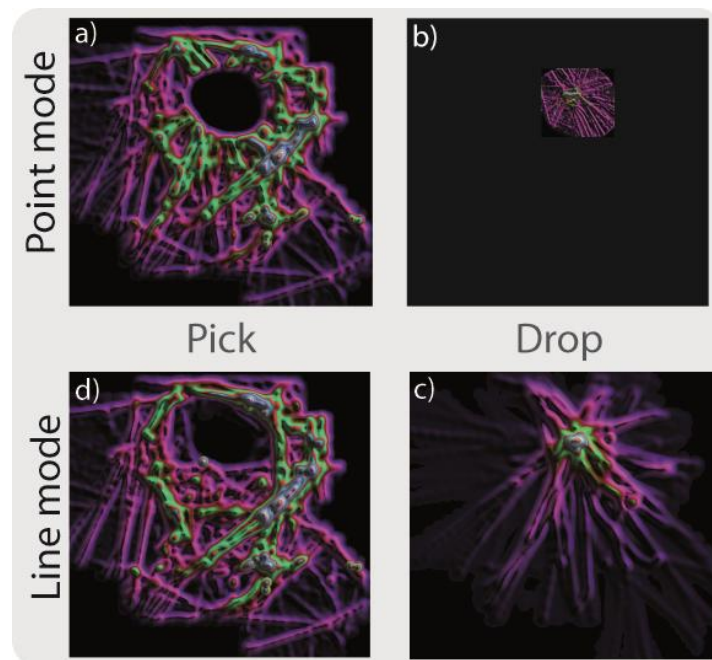
- L'utilisateur peut choisir les dimensions des données à accumuler et peut visualiser en temps réel la carte d'accumulation en choisissant la taille du noyau d'accumulation.
- L'utilisateur peut effectuer des opérations de sélection et d'extraction avec le « Brush, Pick and Drop » et ainsi ajouter ou supprimer des données.
- L'utilisateur peut choisir d'utiliser la carte d'accumulation comme une nouvelle source de données et ainsi affecter des variables visuelles pour afficher les valeurs d'accumulation.

### **1. Interaction avec les cartes d'accumulation**

Nous avons implémenté le paradigme « Pick and Drop » pour les cartes d'accumulation. Ainsi, il est possible de brusher directement dans les cartes d'accumulation pour en sélectionner les données responsables de l'accumulation sélectionné.



L'utilisateur a la possibilité de choisir le mode point ou le mode trajectoire. Dans le mode point, seuls les points qui sont source de l'accumulation sont sélectionnés (images a et b). Dans le mode trajectoire, l'ensemble des points formant la trajectoire est sélectionné (images c et d).



Le « Pick and Drop » des données brushées d'une carte d'accumulation est utilisé dans le processus d'exploration pour plusieurs raisons :

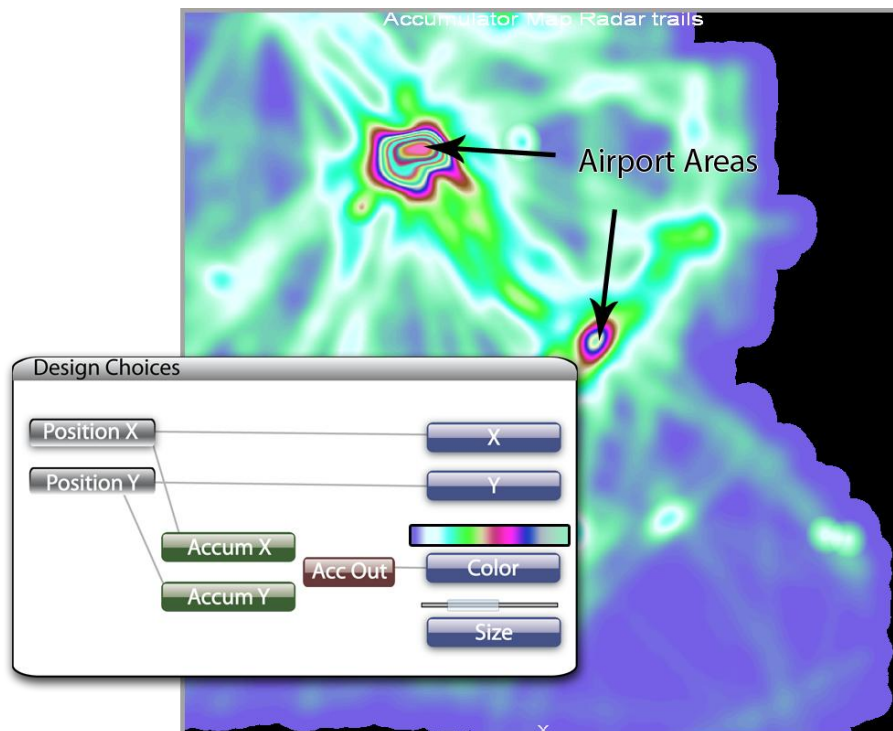
- Il aide à isoler des données pour pouvoir les analyser séparément.
- Dans le mode point ou le mode trajectoire, FromDaDy utilise la plage maximale des variables visuelles. Ainsi, dans le cas de l'utilisation d'un gradient, la valeur minimale d'accumulation est codée par la première couleur du gradient, et la valeur maximale par la dernière couleur du gradient. Cette dynamique permet de mettre à jour de nouvelles zones d'accumulation après chaque action d'isolement de données.



- Dans le mode trajectoire, le brush permet de sélectionner des trajectoires complètes. Ainsi après leur isolation, une nouvelle carte d'accumulation est calculée et de nouvelles zones d'accumulation apparaissent.

## 2. Exemple classique d'accumulation

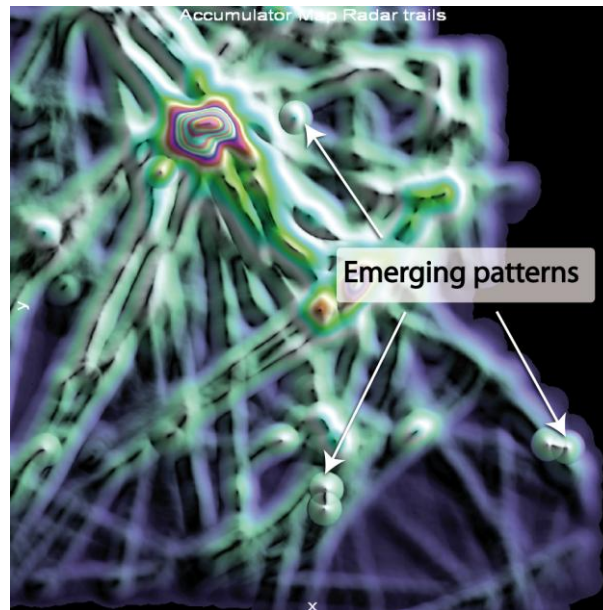
Dans cet exemple, une carte d'accumulation est utilisée pour identifier des patterns et ainsi en trouver l'origine. La base de données utilisée est une journée d'enregistrement du trafic aérien. Chaque enregistrement comporte la position d'un avion, son heure et son indicatif. Lorsque plusieurs avions ont la même position en Latitude/Longitude, il s'opère une accumulation. L'image suivante est le résultat de l'accumulation d'une journée de trafic au dessus de la France en utilisant un Kernel triangulaire :



Dans cette image, l'accumulation s'opère sur les dimensions Position X et Position Y des données. Une échelle de couleur est utilisée pour afficher la valeur d'accumulation. On y voit des zones très denses autour des aéroports (Roissy, Orly, Lyon).

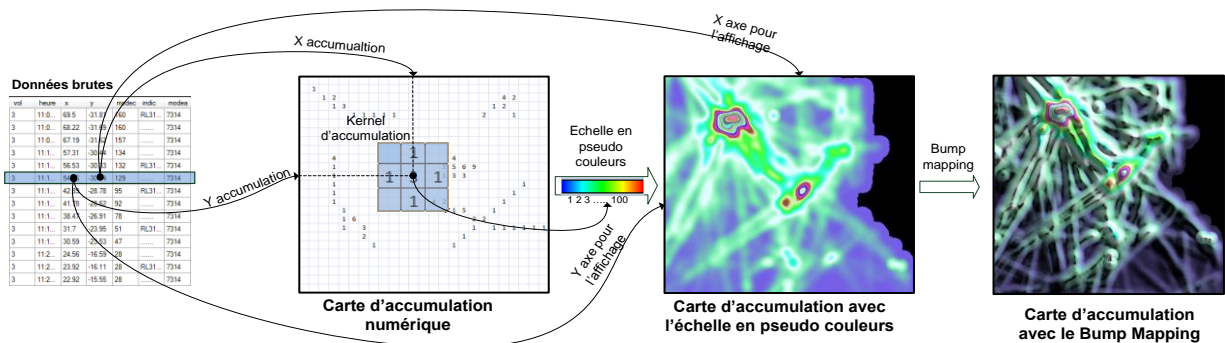
Dans l'image suivante, nous avons utilisé un calcul de la luminosité de chaque point en fonction de la valeur d'accumulation et d'une source d'éclairage.





Des formes circulaires remarquables apparaissent. Elles n'étaient pas aussi visibles sur l'image sans « éclairage ». L'utilisation du calcul de luminosité agit comme un détecteur de contours et renforce alors les « changements brutaux » de valeur d'accumulation. A ce stade, l'utilisateur veut comprendre pourquoi de telles formes apparaissent. Il les sélectionne et les isole en mode points pour pouvoir ainsi explorer directement les données sources de ces accumulations en provoquant l'affichage des données correspondantes sous forme de tableau. Ces points correspondent à des émissions tests des radars pour valider qu'il n'y a pas de dérive dans les coordonnées avions qu'ils émettent. Cette action d'isolement permet aussi d'étendre la plage d'utilisation du gradient pour coder les valeurs d'accumulation. Ces points de test sont émis toute la journée, produisent une forte accumulation et ainsi distordent la répartition des valeurs d'accumulation. En enlevant ces points stationnaires, l'utilisateur augmente la dynamique du gradient.

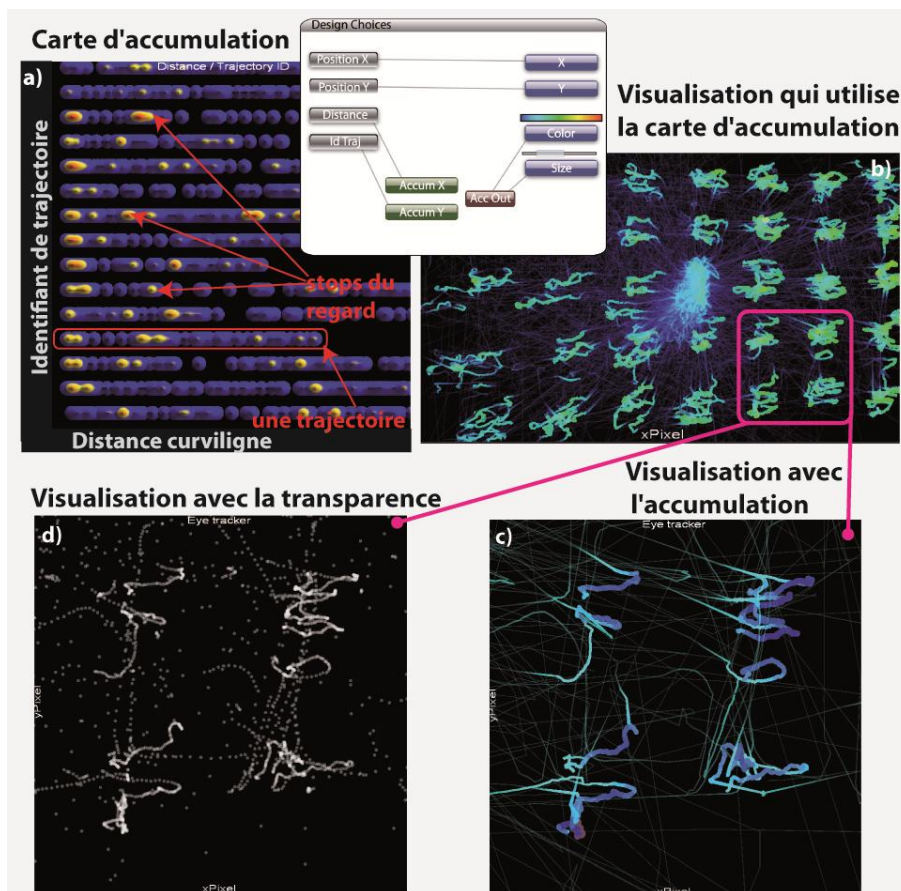
L'image suivante résume l'utilisation et la configuration de design pour la création de la carte d'accumulation d'une journée de trafic.



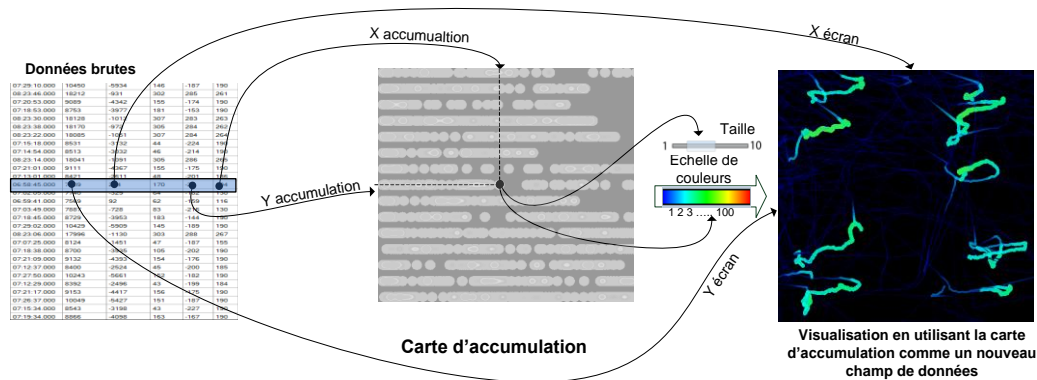
### 3. Utilisation de la carte d'accumulation comme source de données

Dans cet exemple, nous utilisons une base de données contenant les enregistrements d'un eye-tracker. Durant l'enregistrement, l'utilisateur devait regarder au centre de l'écran puis regarder une cible qui apparaissait dans une grille. Chaque séquence a été enregistrée et représente 200 trajectoires (du regard sur la grille) avec 100 000 points au total. Nous avons utilisé la carte d'accumulation pour afficher la vitesse de déplacement du regard et mettre en évidence les points d'arrêt et ainsi permettre d'ordonner les temps d'arrêt du regard. Pour ce faire, nous avons utilisé la configuration d'accumulation suivante : l'identifiant de chaque trajectoire est connecté au Y de l'accumulation, la distance curviligne est connectée au X de l'accumulation. La carte ainsi produite est donnée dans la figure a. La distance curviligne est à 0 quand l'utilisateur commence à quitter le centre de l'écran, puis elle augmente à mesure que l'utilisateur déplace son regard. L'accumulateur permet de mesurer la vitesse de déplacement du regard mais aussi les temps d'arrêt. Les trajectoires sont échantillonnées de façon régulière : ainsi, pour chaque trajectoire, la distance curviligne s'accumule uniquement lorsqu'elle reste identique dans le temps, ce qui correspond à un point de fixation du regard. De la même façon, lorsque la distance curviligne accélère, son accumulation diminue.

La dernière étape pour produire le design est d'utiliser le résultat de la carte d'accumulation pour coder la taille des trajectoires et la couleur.



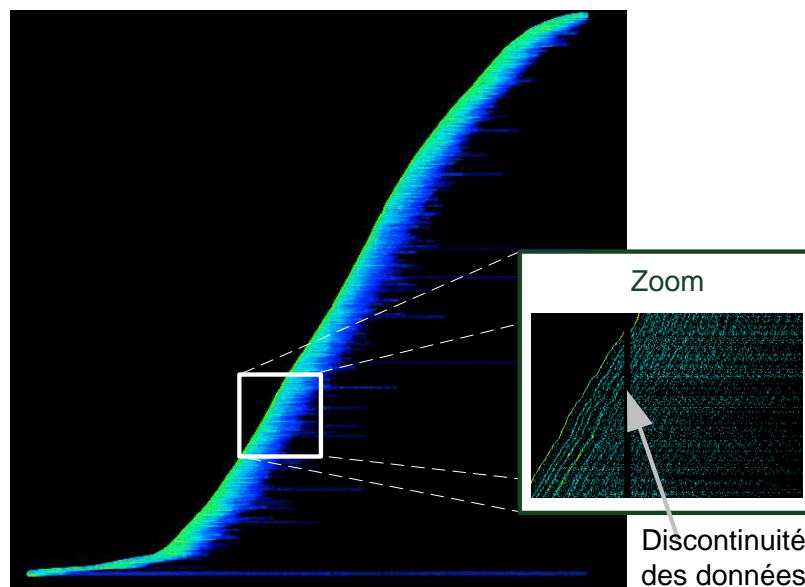
D'une part la carte d'accumulation n'utilise pas les dimensions affichées en X écran et Y écran, d'autre part la carte d'accumulation est utilisée en tant que source de données. L'image suivante résume la configuration à utiliser pour réaliser ce design.



## H. L'accumulation comme un outil d'exploration de données

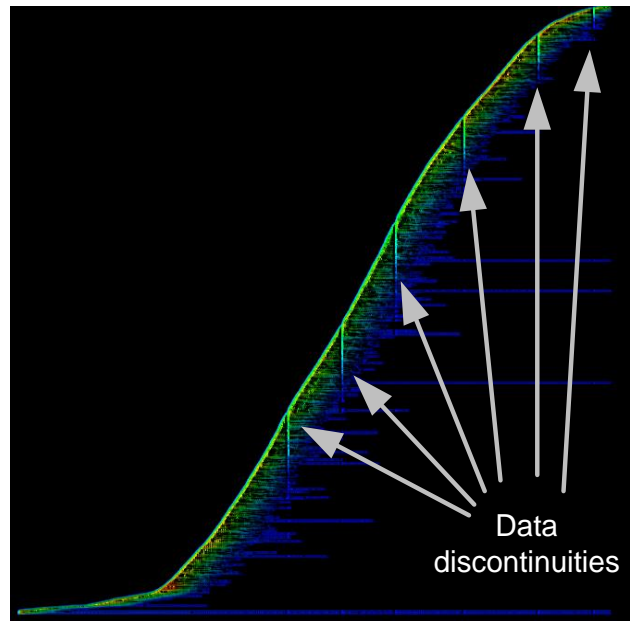
### 1. Détection d'une faille dans le jeu de données

Dans cet exemple, nous allons utiliser l'accumulateur pour détecter un défaut dans le jeu de données. La base de données est une journée d'enregistrement du trafic aérien. Chaque donnée est enregistrée avec un taux d'échantillonnage variant de 3 minutes à 5 minutes. Si on affiche une série temporelle, le nombre trop important de points sur la durée d'une journée ne nous permet pas de mettre en évidence la discontinuité dans le taux d'échantillonnage. L'utilisateur doit alors fortement zoomer sur la visualisation pour faire apparaître les zones de discontinuité.



Il existe une visualisation plus efficace pour afficher cette discontinuité avec l'utilisation de l'accumulation. Comme le taux d'échantillonnage est quasi constant, l'accumulation de la série temporelle devrait être constante elle aussi. La visualisation ainsi produite fait apparaître des stries reflétant à des baisses dans l'apparition des données au cours du

temps. L'utilisateur peut alors voir d'un seul coup toutes les anomalies correspondant à des ruptures dans le taux d'échantillonnage des données.



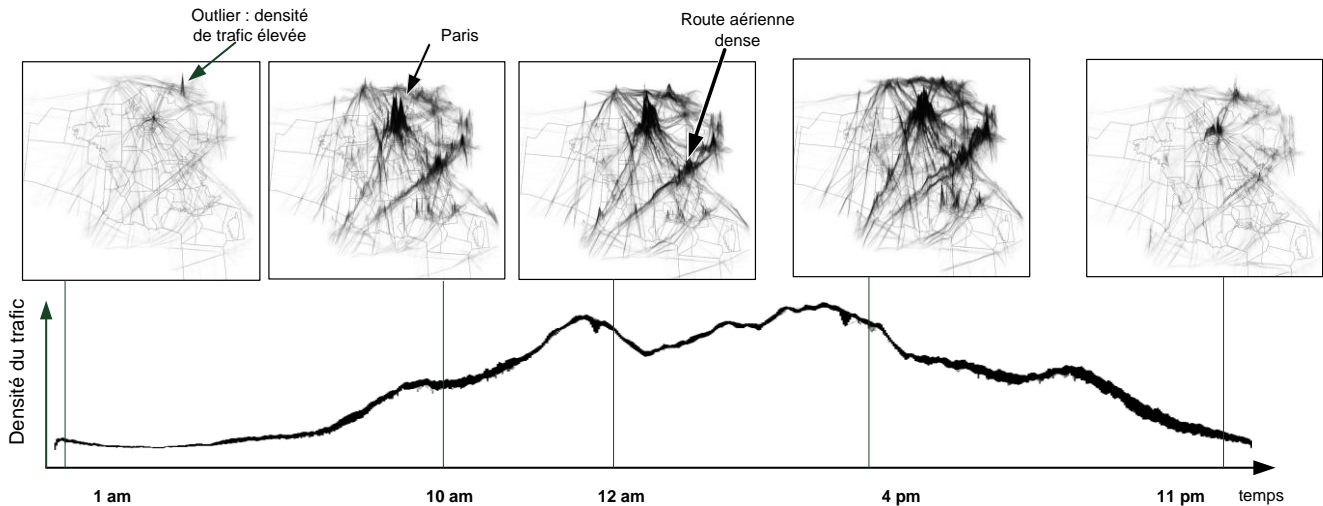
Le design précédent est très efficace car l'utilisation du calcul de luminosité agit comme un détecteur de contour et donc renforce la perception visuelle des rainures dans la série temporelle. D'autres filtrages auraient pu être utilisés, comme par exemple un filtre de Sobel, mais le recours à une lumière virtuelle correspond à un design écologique facilement compréhensible et maniable par un utilisateur.

## 2. Détection d'exceptions dans des flux temporels

Dans cet exemple, nous allons utiliser la fonction d'accumulation avec un filtrage temporel. Ce choix de design permet la création d'une animation qui filtre dans le temps les évolutions de densité de trafic. Une journée standard est organisée comme suit :

- De minuit à 5h, très peu d'avions survolent la France. Ils produisent donc très peu d'accumulation.
- De 5h à 8h, le trafic augmente progressivement et plus particulièrement au dessus de la région parisienne.
- De 8h à 12h, plusieurs routes aériennes sont cycliquement prégnantes.
- De 12h à 13h, de façon globale, le trafic diminue légèrement.
- De 13h à 22h, plusieurs routes aériennes sont cycliquement prégnantes.
- De 22h à 24h, le trafic diminue graduellement jusqu'à produire très peu d'accumulation.

La visualisation suivante résume l'animation de l'accumulation de trafic avec des miniatures prises à des instants remarquables.



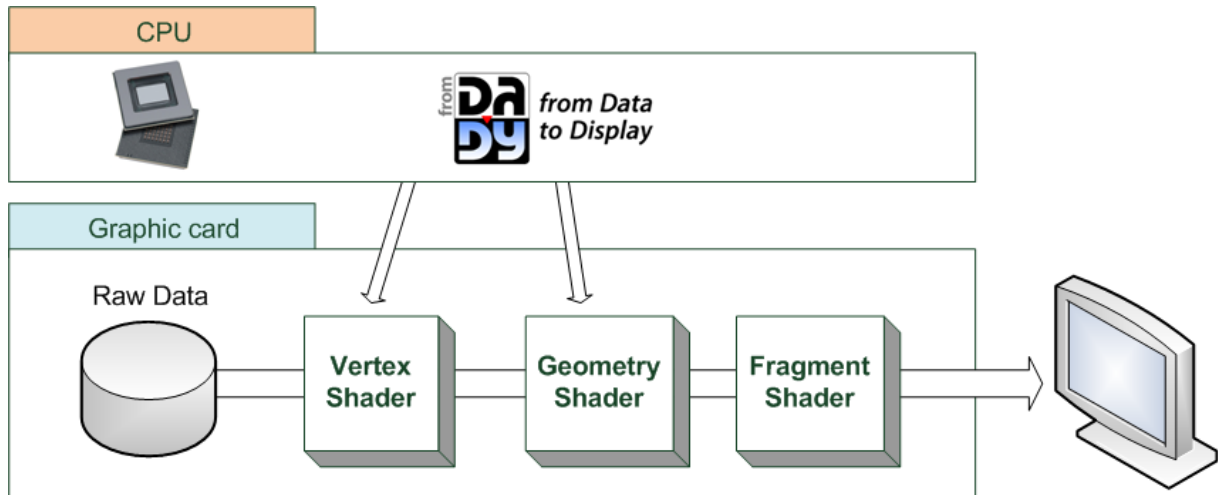
Le résultat de l'animation produite n'est pas surprenant dans le sens où il correspond à une journée type de trafic. Cette animation est facilement compréhensible et permet simplement d'illustrer l'évolution du trafic sur une journée. Elle donne les bases pour effectuer des investigations complémentaires (« Overview first, details on demand » [Shneiderman, 1996]).

Après une analyse plus précise de cette journée de février 2008, l'utilisateur a remarqué une accumulation anormale à 1h. Normalement, très peu d'avions sont censés voler à cette heure. De plus, la zone d'accumulation est réduite à un point au dessus de la Belgique. L'utilisateur utilise alors les outils de sélection pour isoler les trajectoires et constate que 12 avions ont effectué des circuits d'attente pendant deux heures avant d'atterrir, principalement en Belgique. Après recoupement avec des informations météo, il se trouve que ce jour là, des brouillards givrants ont empêché les avions de se poser. Cette exemple montre comment l'animation et l'utilisation de l'accumulation permet de faire de l'exploration de données en mettant en évidence des données anormales.

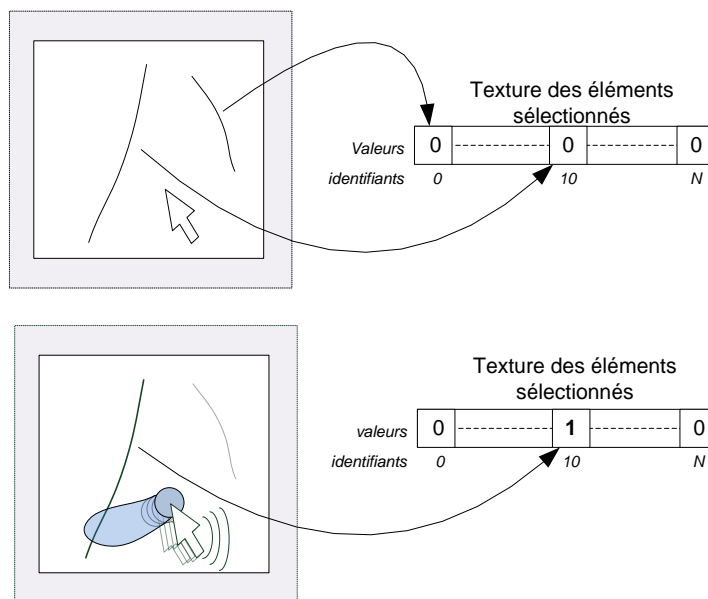
## I. Considérations techniques

FromDaDy a été réalisé avec C# (Framework 3.0) pour l'implémentation de l'interface et DirectX 10 pour la partie de rendu graphique. La réalisation du brush avec 5 millions de points représente un challenge technologique, c'est pourquoi il faut utiliser les capacités des cartes graphiques. FromDaDy utilise les techniques offertes par les Geometry, Vertex et Pixel Shaders, et la technique de rendu dans des textures [Harris, 2005].





Chaque trajectoire a un identifiant unique. Une texture (sauvegardée dans la mémoire de la carte graphique) contient l'état de sélection (une valeur booléenne) de la trajectoire, par défaut initialisé à FAUX (pas de trajectoire sélectionnée). Lorsqu'une trajectoire est brushée, la valeur de cette texture passe à VRAI. Les cartes graphiques utilisent un rendu en parallèle, et non séquentiel, ce qui empêche l'écriture et la lecture en même temps dans une texture : la carte graphique ne peut garantir l'état de la texture qu'une fois la totalité du rendu effectuée. Deux étapes sont donc nécessaires :



Tout d'abord, nous testons les intersections éventuelles entre la zone de brush et les trajectoires. Pour ce faire, nous utilisons le rendu vers une texture. La carte graphique calcule la position sur l'écran de chaque trajectoire. Si le point d'une trajectoire à l'écran doit se dessiner dans une zone de brush (lecture de la texture de brush à l'emplacement prévu du point à dessiner) la carte graphique va dessiner un point (assimilé à une valeur booléenne) dans une texture (la texture des trajectoires sélectionnées) aux coordonnées associées à l'identifiant de l'avion. La deuxième étape consiste à dessiner réellement les points en vérifiant avec la texture précédente si la trajectoire est sélectionnée. Si c'est le cas, un design particulier lui est appliqué (trajectoires en gris). De plus, nous avons aussi implémenté un Geometry Shader qui permet de dessiner les lignes des trajectoires avec une certaine épaisseur. Grâce à toutes ces techniques, FromDaDy peut afficher plus de 5

millions de points avec un taux de rafraîchissement de 20 images par seconde pour un PC de génération 2009 et une carte graphique 8800 GTX.

## J. Conclusion sur FromDaDy

---

FromDaDy utilise une interface minimaliste : une matrice infinie et une fenêtre de design pour les choix de représentation. Les interactions sont elles aussi minimalistes avec le paradigme de « Pick and Drop ». Néanmoins, les combinaisons avec ce paradigme permettent la réalisation de nombreuses tâches : la création et la destruction de vue, la sélection et l'ajustement de sélection, le filtrage d'un sous-ensemble de données, l'application d'opérations booléennes, la création d'étapes importantes pendant le processus d'exploration (story board), l'organisation de son espace de travail.

A travers plusieurs scénarios, nous avons montré comment FromDaDy peut se révéler être un outil efficace pour la création itérative de requêtes avec un grand nombre de données qui contiennent des erreurs et des incertitudes. Ainsi FromDaDy est un outil puissant, précis et flexible avec de nombreuses applications possibles dans le domaine de l'ATC.

FromDaDy n'est pas un nouveau système de visualisation et d'interaction, mais plutôt un ensemble de techniques finement assemblées pour permettre la réalisation d'une tâche précise : l'exploration de trajectoires avions. L'originalité de FromDaDy réside dans l'utilisation de la technique d'étalement des données sur l'espace de travail avec le paradigme de « Pick and Drop ».

L'ajout de la configuration explicite et de la manipulation des cartes d'accumulation a permis d'augmenter les possibilités d'exploration de FromDaDy :

- Détection de patterns comme les points tests radar,
- Détection d'erreurs dans les données comme la rupture de fréquence d'échantillonnage,
- Détection de trajectoires insolites (avions en attente) avec l'utilisation combinée de l'accumulation et de l'animation.

Les cartes d'accumulation produisent des visualisations écologiques car les utilisateurs perçoivent des montagnes et des vallées. Ces indices visuels de haut niveau rendent ce genre de visualisation très efficace pour transmettre de l'information. De plus l'utilisation de l'éclairage permet d'amplifier la détection des contours et de mettre en évidence des éléments remarquables.

FromDaDy a été utilisé de nombreuses fois par des experts du domaine de l'ATC pour analyser des trajectoires d'avions. Nous continuons à faire évoluer ce logiciel avec de nouvelles fonctionnalités et l'exploitation de nouvelles bases de données.







# Conclusion

## Rappel de la problématique

D'une part, nous avons étudié dans cette thèse les problèmes rencontrés par les designers lors de la réalisation de visualisations. L'évaluation des visualisations par les utilisateurs est aujourd'hui une étape incontournable mais limitative, car elle nécessite de disposer d'utilisateurs à chaque itération. Le designer a besoin d'outils permettant d'instrumenter la conception de visualisations pour la rendre plus rapide et plus efficace.

D'autre part, nous avons utilisé nos connaissances pour la production de visualisations efficaces avec les vues métré pour les routes aériennes. Nous avons aussi étudié les problèmes rencontrés lors de l'exploration de grandes quantités de données contenant des erreurs et des incertitudes. L'utilisateur a besoin d'outils de visualisation interactifs pour instrumenter efficacement l'analyse de données.

## La spécificité de notre démarche

Nous avons ancré notre démarche dans le domaine d'application du trafic aérien avec l'étude des visualisations et l'analyse des données de ce domaine. L'état de l'art couvre de nombreux domaines de recherche (perception, caractérisation, évaluation, exploration de données) sur lesquels nous nous sommes appuyés tout au long de cette thèse.

Nous avons rassemblé un ensemble de visualisations du domaine de l'ATC pour couvrir un large spectre de l'espace de design. Nous avons ensuite appliqué les outils de caractérisation issus du domaine de l'InfoVis aux visualisations du domaine du Contrôle Aérien.

Afin d'évaluer la correction de notre analyse, nous avons développé un logiciel qui utilise le modèle de DataFlow et la description d'une visualisation pour recréer les entités visuelles principales qui la composent.

Le logiciel de validation nous a permis d'une part de comprendre le processus de transformation de données vers une visualisation, et d'autre part d'interagir avec les données affichées et ainsi procéder à leur exploration. En effet, ce logiciel est basé sur une visualisation interactive avec des choix de représentations configurables pour amplifier la cognition. C'est ainsi que notre démarche de validation s'est transformée en réalisation d'un logiciel capable de faire de l'exploration interactive de données.

La caractérisation est la pierre angulaire de nos travaux de recherche présentés dans cette thèse. Premièrement, cet outil nous a permis de définir et d'exploiter avec précision les détails des visualisations. Deuxièmement, c'est l'étude de la caractérisation des cartes de type métré qui a permis leur transposition dans le domaine du Contrôle Aérien avec l'affichage schématique des routes aériennes. Enfin, l'utilisateur configure le visuel de FromDaDy en s'appuyant sur le modèle de DataFlow et sur sa caractérisation.

### **Les apports essentiels de notre travail**

Tout d'abord, nous avons identifié en quoi la caractérisation des visualisations peut être utile pour les concepteurs. Une telle caractérisation pourrait permettre :

- de fournir un langage commun,
- de comparer les visualisations,
- de décrire un espace de design délimité.

Nous avons identifié que cette caractérisation est composée de deux éléments : son contenu et sa mise en forme. La découverte de l'importance de la mise en forme au même niveau d'importance que la description est un apport de l'étude de l'état de l'art. Cette dissociation est essentielle pour l'exploitation de la caractérisation. Ainsi le contenu décrit les informations discriminantes de la visualisation. La mise en forme permet de structurer les données caractéristiques des visualisations et ainsi faciliter leur exploitation. Par exemple, le contenu de la description peut être réalisé au moyen d'étude avec la théorie des échelles, la mise en forme avec une structure sous forme de tableau.

Nous avons caractérisé des visualisations issues de la classe de supervision, utilisées par les contrôleurs aériens. Notre démarche nous a permis de valider que les modèles de caractérisation de l'InfoVis sont, dans une certaine mesure, applicables à cette classe de visualisations. Une contribution de cette thèse est l'utilisation des travaux de C&M pour caractériser les visualisations et l'utilisation de ces caractérisations pour identifier et corriger des incohérences dans les designs.

Nous avons identifié les limites du modèle de C&M avec la caractérisation de la seule fonction principale des visualisations et l'impossibilité de caractériser les données émergentes. Nous avons donné une définition et étudié la notion d'émergence, et nous avons discuté de son utilisation dans les visualisations.

Nous avons proposé une méthode ad-hoc pour caractériser les visualisations avec une description exhaustive des informations qu'elles contiennent. L'étude approfondie des designs des comètes radar a permis de mieux comprendre leurs différences.

Ensuite, nous avons exploré le domaine de la visualisation et de l'interaction avec de grandes quantités d'information. Notre apport est la réalisation d'un paradigme d'interaction efficace : le « brush, pick and drop » qui permet d'instrumenter la manipulation de trajectoires. Une autre contribution est l'utilisation de l'étalement des données dans une matrice infinie de vues. Les trajectoires ne sont pas dupliquées mais seulement déplacées, ce qui permet à l'utilisateur d'organiser efficacement son espace de travail : chaque vue est une sous-partie des données initiales.

Nous avons aussi contribué à étendre les manipulations de données aux cartes d'accumulation. Ainsi, l'utilisateur peut explorer interactivement des données accumulées, les déplacer, changer leur représentation et les utiliser comme nouvelle source de données, c'est-à-dire un nouveau champ disponible. L'utilisateur peut expliciter les choix de design pour la réalisation de ces cartes avec la définition de leur caractérisation.

La conception de FromDaDy est basée sur l'utilisation d'un ensemble restreint de fonctionnalités en maximisant leur pouvoir d'expression. Ainsi, le système est non-modal (pas de menu, pas de fenêtre de configuration bloquante), toutes les manipulations sont réalisées directement et rapidement. De plus, FromDaDy affiche des designs écologiques (cartes d'accumulation).

FromDaDy utilise les fonctionnalités avancées des cartes graphiques ce qui permet de le rendre très efficace en terme de réactivité lors de la manipulation de grandes quantités de données. L'exploration de trajectoires est rendue plus efficace et plus rapide et ainsi augmente la performance de l'utilisateur. Ce logiciel est régulièrement utilisé au sein des laboratoires de recherche de l'Aviation Civile pour la manipulation de données diverses : les trajectoires d'avions et de drones, les résultats d'algorithmes de prévision de trajectoires, les maillages des connexions entre avions. La réalisation d'un logiciel opérationnel est une contribution importante pour notre laboratoire, mais est aussi un support à notre recherche.

### **Perspectives**

Notre démarche est une première étape dans le processus de caractérisation et d'exploitation des dimensions discriminantes des visualisations. La description exhaustive est une étape importante mais la quantité d'informations qu'elle engendre (même structurée) nuit à son utilisation. Nous pensons donc filtrer les informations caractéristiques des visualisations en fonction des tâches souhaitées par le designer. Cette approche nécessite de décrire les tâches en s'appuyant, par exemple, sur les travaux de Casner [Casner, 1991].

Nous pensons continuer à explorer la caractérisation de visualisation en utilisant d'autres critères qualitatifs comme le nombre d'informations émergentes [HURTER 09], ou encore des critères économiques [Van Wijk, 2005]. Cette démarche va nous permettre de réaliser une taxonomie des visualisations de l'ATC rendant alors possible la structuration de l'espace de design et, par extension, permettre de trouver des dimensions non utilisées pour la réalisation de nouvelles visualisations. Ces axes de recherches se concentrent sur une problématique plus générale : trouver les bonnes dimensions de caractérisation.

Notre deuxième axe de recherche concernant l'exploration de données nous a permis de valider l'efficacité des outils de visualisation et d'interaction avec des données ATC. Nous avons déjà procédé à de nombreuses autres explorations de données dans les domaines suivants : physiologique, mathématique, base de données... Une volonté commune des utilisateurs s'est dégagée de ces explorations : la manipulation de données est très efficace et la notion de conservation du point de focus de l'utilisateur est primordiale lors des changements de design de la visualisation. La notion de transition douce et contrôlée est la pierre angulaire d'une visualisation efficace. Nous pensons donc continuer dans cette direction avec une utilisation plus marquée du contrôle de l'utilisateur dans le processus de transition entre différents designs.

Nous espérons qu'au travers de cette thèse d'autres recherches seront menées sur caractérisation de visualisations afin de contribuer aux efforts pour la production efficace de visualisations interactives.



---

## Références



### Visualisation des connexions dans le cerveau.

*Maarten H. Everts, Henk Bekker, Jos B. T. M. Roerdink, and Tobias Isenberg (2009) Depth-Dependent Halos: Illustrative Rendering of Dense Line Data. IEEE Transactions on Visualization and Computer Graphics, 15(6):1299–1306, November/December 2009. Best Paper Award at IEEE Visualization 2009.*

- [Ahlberg et al., 1992]  
Ahlberg, C., Williamson, C., and Shneiderman, B. 1992. **Dynamic queries for information exploration: an implementation and evaluation.** Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Monterey, California, United States, May 03 - 07, 1992). CHI '92. ACM, New York.
- [Ahlberg & Schneiderman, 1994]  
C. Ahlberg and B. Shneiderman, "**Visual Information Seeking using the FilmFinder,**" presented at CHI'94, 1994.
- [Ahlberg, 1996]  
Ahlberg, C. 1996. **Spotfire: an information exploration environment.** *SIGMOD Rec.* 25, 4 (Dec. 1996).
- [Anderson, 1935]  
Anderson, E. 1935. **The irises of the Gaspé Peninsula, Bulletin of the American Iris Society,** 59.
- [Appert et al., 2004]  
Appert, C., Beaudouin-Lafon, M., Mackay, W., **Context matters: Evaluating interaction techniques with CIS model.** HCI 2004.
- [Athènes et al. 2000]  
Athènes, S., Chatty, S., Bustico, A., **Human factors in ATC alarms and notifications design: experimental evaluation.** In: Proc. ATM'2000 R&D seminar (2000).
- [Bartram et al., 2001]  
Bartram, L., Ware, C., Calvert, T., 2001. **Moving icons: detection, distraction and task.** In: Hirose, M. (Ed.), Proceedings of the IFIP International Conference on Human-Computer Interaction (INTERACT) '01, IOS Press, Tokyo.
- [Baudel, 2002]  
Baudel, T. 2002. **Visualisations compactes: une approche déclarative pour la visualisation d'information.** Proceedings of the 14th French-Speaking Conference on Human-Computer interaction (Conférence Francophone Sur L'interaction Homme-Machine). IHM '02, vol. 32. ACM.
- [Baudel, 2007]  
Baudel, T. 2004. **Browsing through an information visualization design space.** In CHI'04 Extended Abstracts on Human Factors in Computing Systems CHI '04. ACM, New York.
- [Baudel, 2002]  
Baudel, T., **Canonical Representation of Data-Linear Visualizations and its Applications,** ILOG Technical report 02-002. 2002.
- [Baudisch et al., 2006]  
Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, A., Zhao, S., Ramos, G., **Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects.** UIST 2006.
- [Becker & Cleveland, 1987]  
Becker, R. A., Cleveland, W. S., **Brushing scatterplots.** *Technometrics* (1987), Vol. 29, No. 2. (1987).
- [Bekos et al., 2007]  
Bekos M., Kaufmann M., Potika K., Symvonis A., **Line Crossing Minimization on Metro Maps.** Proc. of Graph Drawing 2007.
- [Benhacene, 2002]  
Benhacene, R., **A Vertical Image as a means to improve air traffic control in E-TMA.** USA, DASC California, 2002.
- [Berlin & Kay, 1969]  
Berlin, B., Kay, P., **Basic color terms : their universality and evolution.** Berkeley; Oxford: Univer-sity of California Press (1969).

- 
- [Bertin, 1983]  
Bertin, J. 1983. **Semiology of Graphics**. University of Wisconsin Press.
- [Bezerianos et al., 2006]  
Bezerianos, A., Dragicevic, P., Balakrishnan, R., **Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays**. UIST 2006.
- [Callahan et al., 1988]  
Callahan, J., Hopkins, D., Weiser, M., Shneiderman, B. 1988. **An empirical comparison of pie vs. linear menus**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '88. ACM.
- [Card et al., 1980]  
Card, S.K., Moran, W.P., Newell, A. **The keystroke-level model of user performance time with interactive systems**. Communication of the ACM 23.1980.
- [Card et al., 1986]  
Card, S.K., Moran, W.P., Newell, A. **The model human processor: an engineering model for human performance**. Handbook of Perception and Human Performance, Wiley 1986.
- [Card & Mackinlay 1997]  
Card, S.K., Mackinlay, J.D., **The Structure of the Information Visualization Design Space**. In Proc. Information Visualization Symposium '97.
- [Card et al., 1999]  
Card, S.K., Mackinlay, J.D., Shneiderman, B., Eds. 1999 **Readings in Information Visualization: Using Vision to Think**. Morgan Kaufmann Publishers Inc.
- [Casner, 1991]  
Casner, S. M. 1991. **Task-analytic approach to the automated design of graphic presentations**. ACM Trans. Graph. 10, 2 (Apr. 1991).
- [Chambers, 1997]  
Chambers, J.M. (1977). **Computation Methods for Data Analysis**. New York: John Wiley & Sons.
- [Chambers et al., 1983]  
Chambers, J.M., Cleveland, W.S., Tukey, P.A., **Graphical Methods for Data Analysis**, Wadsworth, 1983.
- [Chi, 2000] [CHI 00]  
Chi, E. H. 2000. **A Taxonomy of Visualization Techniques Using the Data State Reference Model**. In *Proceedings of the IEEE Symposium on information Vizualization 2000*. INFOVIS. IEEE Computer Society, Washington, DC.
- [CIE CAM 02]  
**A Colour Appearance Model for Colour Management Systems: CIE CAM 2002**, CIE 159.
- [Cleveland, 1985]  
Cleveland, W. S. 1985 **The Elements of Graphing Data**. Wadsworth Publ. Co.
- [Cleveland & McGill, 1988]  
Cleveland, W. S, McGill, M.E. 1988 **Dynamic Graphics for Statistics**. Belmont, CA: Wadsworth.
- [Cleveland, 1995]  
Cleveland, W.S. 1995 **Visualizing Data**. Summit, NJ: Hobart Press.
- [Conversy et al., 2003]  
Conversy, S., Roussel, N., Hansen, H., Evans, H., Beaudouin-Lafon, M., Mackay.W., **Sharing daily-life images with videoProbe**. Proceedings of the 15th French-Speaking Conference IHM 2003.
- [Conversy et al., 2007]  
Conversy, S., Lemort, A., Valès, S., **MAMMI WP1: Review of existing systems**. Technical report, T06-043, Eurocontrol, 2007.

- [Conversy et al., 2010]  
Conversy, S., Hurter, C., Chatty, S. **A Descriptive Model of Visual Scanning**. Proceedings of the 2010 Conference on Beyond Time and Errors: Novel Evaluation Methods For Information Visualization (to appear) (long article at BELIV). ACM Press, 2010.
- [Cook & Weisberg, 1994]  
Cook, R.D., Weisberg, S. 1994, **An Introduction to Regression Graphics**, New York: John Wiley & Sons.
- [Derthick et al., 1997]  
Derthick, M., Kolojejchick, J., and Roth, S. F. 1997. **An interactive visual query environment for exploring data**. Proceedings of the 10th Annual ACM Symposium on User interface Software and Technology. UIST '97. ACM, New York, NY.
- [Ellis et al., 1990]  
Ellis, C., Gibbs, S.J., Rein, G, **Design and Use of a Group Editor**. Engineering for Human Computer Interaction, G. Cockton, editor. North Holland, Amsterdam, 1990.
- [Elmqvist et al., 2008]  
Elmqvist, N.; Dragicevic, P.; Fekete, J.-D., **Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation Visualization**, *IEEE Transactions on* Volume 14, Issue 6, Nov.-Dec. 2008.
- [Falmagne, 1985]  
Falmagne, J-C, **Elements of Psychophysical Theory**. Oxford: Oxford University Press. 1985.
- [Fekete, 2004]  
Fekete, JD. 2004. **The InfoVis Toolkit**. Proceedings of the IEEE Symposium on information Visualization 2004. INFOVIS. IEEE Computer Society, Washington, DC.
- [Garland, 1994]  
Garland K.: **Mr Beck's Underground Map**, Capital Transportation Publishing, 1994.
- [Gibson, 1977]  
Gibson, J.J., **The theory of affordances**. Perceiving, Acting, and Knowing: Toward an Ecological Psychology. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.
- [Gibson, 1979]  
Gibson, J.J., **The Ecological Approach to Visual Perception**. Houghton Mifflin, Boston. 1979.
- [Harris, 2005]  
Harris, M. 2005. **Mapping computational concepts to GPUs**. In ACM SIGGRAPH 2005 Courses. J. Fujii, Ed. SIGGRAPH '05. ACM, New York, NY, 2005.
- [Harrower et al., 2003]  
Harrower, M. and Brewer, C.A., ColorBrewer.org: an online tool for selecting colour schemes for maps. *Cartogr. J.* v40 i1. 27-37.
- [Healea, 1992]  
Healey ,C.G., **Visualization Of Multivariate Data Using Preattentive Processing** , Master's Thesis, 1992 , Department of Computer Science, University of British Columbia.
- [Heer & Robertson, 2007]  
Heer, J. and Robertson, G. 2007. **Animated Transitions in Statistical Data Graphics**. *IEEE Transactions on Visualization and Computer Graphics* 13, 2007.
- [Hering, 2004]  
Hering, H., **Vital: An advanced time-based tool for the future 4D ATM environment**. Proceedings of 23<sup>rd</sup> DASC Conference. Salt Lake City, USA 2004.
- [Hochheiser & Shneiderman, 2004]  
Hochheiser, H. and Shneiderman, B. 2004. **Dynamic query tools for time series data sets: timebox widgets for interactive exploration**. *Information Visualization* 2004.



---

[HURTER IHM 07]

Hurter, C., Conversy, S., **Extension d'un modèle de visualisation pour la caractérisation d'interfaces graphiques dynamiques**, 19ème Conférence Francophone sur l'Interaction Homme-Machine, IHM 07, IRCAM Paris, ACM Press.

[HURTER DSVIS 08]

Hurter, C. and Conversy, S. 2008. **Towards Characterizing Visualizations**. In interactive Systems. Design, Specification, and Verification: 15th international Workshop, DSV-IS 2008. Springer-Verlag.

[HURTER ICRAT 08]

Hurter, C., Conversy, S., Kapp, V., **An Infovis approach to compare ATC comets**. 3<sup>rd</sup> International Conference on Research in Air Transportation. ICRAT 2008 Fairfax Virginia USA.

[HURTER CHI 09]

Hurter, C., Conversy, S., Vinot, J.L., **Temporal data visualizations for Air Traffic Controllers**. Interacting with temporal data. CHI 09 Workshop.

[HURTER INFOVIS 09]

Hurter, C., Tissoires, B., and Conversy, S. 2009. **FromDaDy: Spreading Aircraft Trajectories Across Views to Support Iterative Queries**. IEEE Transactions on Visualization and Computer Graphics 2009.

[HURTER IHM 09]

Hurter, C., Conversy, S., Vinot, J., and Jestin, Y. 2009. **Représentations écologiques de données temporelles: exemples et apports**. Proceedings of the 21st international Conference on Association Francophone D'interaction Homme-Machine (2009). IHM '09. ACM, New York, NY.

[HURTER IHM short 09]

Hurter, C., Serrurier, M., Gilles, T., and Vinot, J. 2009. **Génération et placement de couleurs sur une vue de type métro**. Proceedings of the 21st international Conference on Association Francophone D'interaction Homme-Machine. IHM '09. ACM, New York, NY.

[HURTER AVI 10]

Hurter, C., Serrurier, M., Gilles, T., and Vinot, J. 2010. **An automatic generation of metro-like maps to display Flight Routes for Air Traffic Controllers: structure and color optimization**. Conference Proceedings of AVI'10, Advanced Visual Interfaces, Roma, Italy may 2010, ACM Press.

[Inselberg & Dimsdale, 1990]

Inselberg, A., Dimsdale, B., (1990), **Parallel coordinates: a tool for visualizing multi-dimensional geometry**, Proceedings of the First IEEE Conference on Visualization, San Francisco, CA.

[Ishizaki, 1995]

Ishizaki, S. 1995. **Color adaptive graphics: what you see in your color palette isn't what you get**. Conference Companion on Human Factors in Computing Systems. 1995. CHI '95. ACM, New York, NY.

[Johnson & Shneiderman, 1991]

Johnson, B., Shneiderman, B. 1991. **Tree-Maps: a space-filling approach to the visualization of hierarchical information structures**. *Proceedings of the 2nd Conference on Visualization '91*. Visualization. IEEE Computer Society Press, Los Alamitos, CA.

[Keim et al., 2006]

Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. 2006. **Challenges in Visual Data Analysis**. Proceedings of the Conference on information Visualization 2006. IV. IEEE Computer Society, Washington, DC.

[Kirkpatrick et al., 1983]

Kirkpatrick S., Gelatt C. D., Vecchi M. P., **Optimization by simulated annealing**. Science 220, 671-680, 1983.

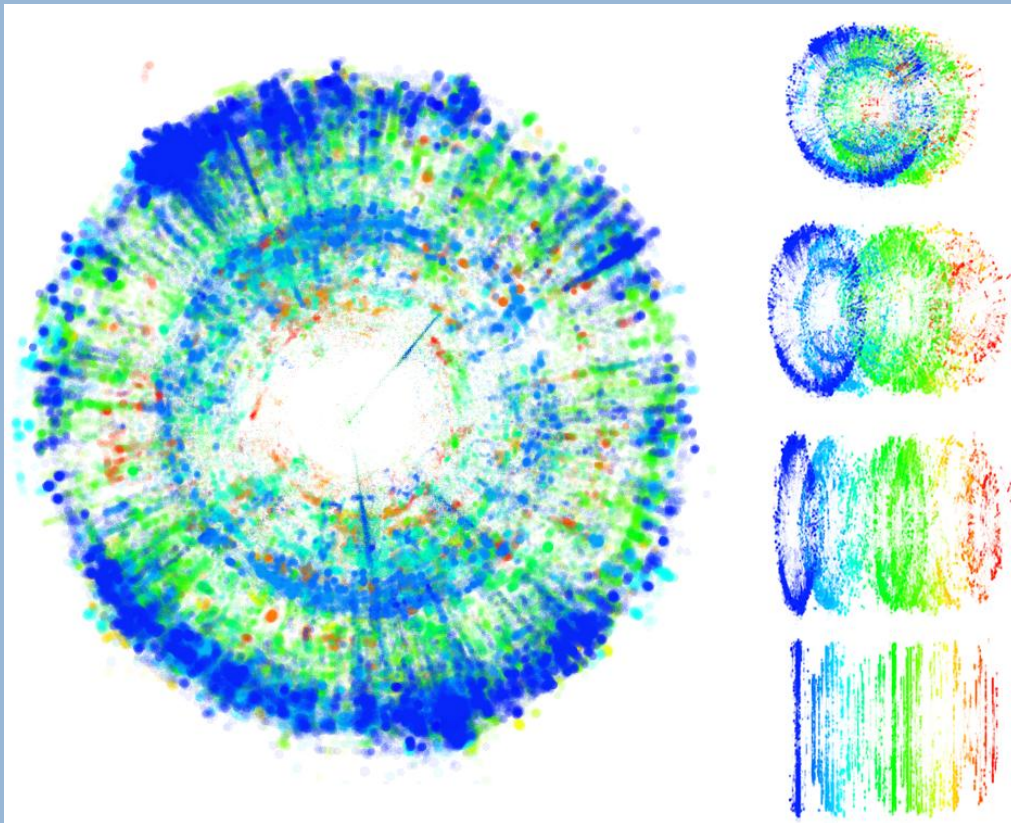
- [Koffka, 1935]  
Koffka, K., **Principles of Gestalt psychology**. Routledge 1935.
- [Kosslyn, 1994]  
Kosslyn, S.M., **Elements of Graph Design**, New York : W.H. Freeman 1994.
- [Kurtenbach & Buxton, 1991]  
Kurtenbach, G., Buxton, B., 1991. **Issues in combining marking and direct manipulation techniques**. Proceedings of the 4th Annual ACM Symposium on User interface Software and Technology. UIST '91. ACM, New York, NY, 137-144.
- [Lejeune & Wearden, 2009]  
Lejeune H., Wearden, J.H., **Vierordt's The Experimental Study of the Time Sense (1868) and its legacy**, in European Journal of Cognitive Psychology January 2009.
- [Levkowitz, 1992]  
Levkowitz, H., Herman, G. T. 1992. **Color Scales for Image Data**. IEEE Comput. Graph. 1992.
- [Livny et al., 1997]  
Livny, M., Ramakrishnan, R., Beyer, K., Chen, G., Donjerkovic, D., Lawande, S., Myllymaki, J., and Wenger, K. 1997. **DEVis: integrated querying and visual exploration of large datasets**. *SIGMOD Rec.* 26, 2 .1997.
- [MacEachren, 1995]  
MacEachren, A. M. 1995. **How Maps Work: Representation, Visualization and Design**. Guilford Press.
- [Mackinlay, 1986]  
Mackinlay, J. 1986. **Automating the design of graphical presentations of relational information**. ACM Trans. Graph. 5, 2 1986.
- [Martin et al., 1995]  
Martin, A. R. and Ward, M. O. 1995. **High Dimensional Brushing for Interactive Exploration of Multivariate Data**. In *Proceedings of the 6th Conference on Visualization '95*. IEEE Visualization. IEEE Computer Society, Washington, DC.
- [Merrick & Gudmundsson, 2006]  
Merrick D., Gudmundsson J., **Path Simplification for Metro Map Layout**. Proc in Graph Drawing 2006.
- [Miller, 1956]  
Miller, A., The **Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information**. The Psychological Review 1956.
- [Moretti & Lyons, 2005]  
Moretti, G.S., Lyons, P.J., **Controlling the complexity of grouped items in colour interfaces**. Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural, ACM (2005).
- [Morrison, 1974]  
Morrison, J.L., **A Theoretical framework for cartographic generalization with the emphasis on the process of symbolization**. International Yearbook of Cartography, 14, 1974.
- [Munsell, 1912]  
Munsell, A. H., **A Pigment color System and Notation**. The American Journal of Psychology. 1912.
- [Nobel & Sperandio, 1973]  
Nobel, J., Sperandio, J., **Etude expérimentale du strip "base temp" à l'usage des centres des control régionaux**. Centre d'Etude de la Navigation Aérienne, France, 1973.
- [Nöllenburg & Wolf, 2006]  
Nöllenburg, N., Wolf, A., **A mixed-integer program for drawing high-quality metro maps**. Proceedings. 13th Internat. Sympos. Graph Drawing (GD'05), Springer-Verlag, 2006.

- 
- [Norman & Thrower, 1969]  
Norman, J., Thrower, W., Edmond **Halley as a Thematic Geo-Cartographer**, Annals of the Association of American Geographers, Vol. 59, 1969.
- [Norman, 1995]  
Norman, D. A. 1995. **The psychology of everyday things**. In Human-Computer interaction: Toward the Year 2000, Eds. Morgan Kaufmann Publishers, San Francisco, CA.
- [Párraga et al., 2008]  
Párraga, A. Benavente, R Vanrell, M. and Baldrich, R., **Modelling inter-colour regions of Colour Naming Space**. Proceedings on Color in Graphics, Imaging and Vision CGIV'08.
- [Plaisant, 2004]  
Plaisant, C. 2004. **The challenge of information visualization evaluation**. Proceedings of the Working Conference on Advanced Visual interfaces. AVI '04. ACM, New York, NY.
- [Rao & Card, 1994]  
Rao, R., Card, S.K., **The Table Lens: Merging graphical and symbolic representations in an interactive focus plus context visualization for tabular information**, Proceedings. CCHI '94 ACM, 1994.
- [Rheingans, 1999]  
Rheingans, P., **Task-based Color Scale Design**. Proceedings of Applied Image and Pattern Recognition '99, SPIE, 1999.
- [Rekimoto, 1997]  
Rekimoto, **Pick-and-drop: a direct manipulation technique for multiple computer environments**. *Proceedings of the 10th Annual ACM Symposium on User interface Software and Technology*. UIST '97. ACM, New York, NY, 1997.
- [Rhyne et al., 2003]  
Rhyne, T., Tory, M., Munzner, T., Ward, M., Johnson, C., and Laidlaw, D. H. 2003. **Information and Scientific Visualization: Separate but Equal or Happy Together at Last**. Proceedings of the 14th IEEE Visualization 2003 (Vis'03) IEEE Visualization., Washington, DC.
- [Robertson et al., 1991]  
Robertson, G.G., Mackinlay, J.D., Card, S.K. (1991), **Cone Trees: animated 3D visualizations of hierarchical information**, Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, New Orleans, Louisiana, United States, ACM Press.
- [Robertson & O'Callaghan, 1986]  
Robertson, P., O'Callaghan, J. 1986. **The Generation of Color Sequences for Univariate and Bivariate Mapping**. IEEE Comput. Graph. Appl. 6, 2 . 1986.
- [Robertson & De Ferrari, 1994]  
Robertson, P., De Ferrari, L., **Systematic Approaches to Visualization: Is a Reference Model Needed?** Scientific Visualization, 1994, Advances and Challenges, Ed: L. Rosenblum, R.A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, D. Thalmann , Academic Press.
- [Roth et al., 1994]  
Roth, S. F., Kolojejchick, J., Mattis, J., Goldstein, J. 1994. **Interactive graphic design using automatic presentation knowledge**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating interdependence. CHI '94. ACM, New York, NY.
- [Shiffman, 1990]  
Shiffman, H.R., **Sensation and Perception: An Integrated Approach**. New York: John Wiley & Sons. 1990.
- [Schlienger et al., 2007]  
Schlienger, C., Conversy, S., Chatty, S., Anquetil, M., Mertz, C., **Improving Users**

- Comprehension of Changes with Animation and Sound: an Empirical Assessment.** Proceedings of Interact 2007, Lecture Notes in Computer Science. Springer Verlag, 2007.
- [Shneiderman, 1987]  
Shneiderman, B., **Direct manipulation: A step beyond programming languages. In Human-Computer interaction: A Multidisciplinary Approach**, R. M. Baecker, Ed. Morgan Kaufmann Publishers, San Francisco, CA, 1987.
- [Shneiderman, 1996]  
Shneiderman, B. **The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations**, CS-TR-3665 July 1996.
- [Silverman, 1986]  
Silverman, B.W, **Kernel Density Estimation Using the Fast Fourier Transform Applied Statistics**, Chapman and Hall, 1986.
- [Stevens, 1946]  
Stevens, S.S., **On the theory of scales of measurement.** Science,1946.
- [Stevens, 1961]  
Stevens, S.S., **The psychophysics of sensory function.** *Sensory Communication*, ed. W.A. Rosenblith. MIT Press, Cambridge, MA.
- [Stevens, 1985]  
Stevens, S.S., **Psychophysics: Introduction to its Perceptual, Neural, and Social Prospects.** New Brunswick, NJ: Transaction Books 1985.
- [Stole & Hanrahan, 2000]  
Stolte, C., Hanrahan, P., **Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases.** *Proceedings of the IEEE Symposium on information Vizualization 2000.* INFOVIS. IEEE Computer Society, Washington, DC.
- [Stott & Rodgers, 2004]  
Stott, J. M. and Rodgers, P., **Metro Map Layout Using Multicriteria Optimization.** Proceedings of the Information Visualisation International Conference. IV. IEEE Computer Society, 2004.
- [Swayne et al., 2003]  
Swayne, D. F., Lang, D. T., Buja, A., Cook, D., **GGobi: evolving from XGobi into an extensible framework for interactive data visualization.** Computer. Statistic. Data Analysis. 43, 2003.
- [Tabart et al., 2007]  
Tabart, G., Athènes, S., Conversy, S., Vinot, J., **Effets des paramètres graphiques sur la perception visuelle: expérimentations sur la forme, la surface, l'orientation des objets et la définition des écrans.** Proceedings of Association Francophone D'interaction Homme-Machine IHM '07, vol. 397. ACM, New York, NY.
- [Thomas & Cook, 2006]  
Thomas, J. J., Cook, K.A. **Illuminating the Path: The Research and Development Agenda for Visual Analytics**, IEEE Computer Society, 2006.
- [Treinish, 1994]  
Treinish, L., **Ozone animation** IBM, 1994.
- [Treisman, 1985]  
Treisman, A., Preattentive processing in vision. Computer. Vision Graphic. Image Processessing. 31, 1985.
- [Treisman, 1988]  
Treisman, A., Gormican, S., **Feature analysis in early vision: Evidence from search asymmetries.** Psychological Review 95, 1988.
- [Tufte, 1983]  
Tufte, E.R., **The Visual Display of Quantitative Information**, Graphics Press, Chesire, Connecticut, 1983.

- 
- [Tufte, 1990]  
Tufte, E.R., **Envisioning Data**, Graphics Press, Chesire, Connecticut, 1990.
- [Tufte, 1997]  
Tufte, E.R., **Visual Explanations**, Graphics Press, Chesire, Connecticut, 1997.
- [Tukey, 1977]  
Tukey, J.W., **Exploratory Data Analysis**. Reading, MA: Addison-Wesley, 1977.
- [Tweedie, 1997]  
Tweedie, L.. Characterizing **interactive externalizations**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '97. ACM, New York, NY, 1997.
- [Ward, 1994]  
Ward, M. O., **XmdvTool: integrating multiple methods for visualizing multivariate data**. *Proceedings of the Conference on Visualization '94* IEEE Visualization. IEEE Computer Society Press, 1994.
- [Ware, 2000]  
Ware, C., **Information Visualization: Perception for Design**, Morgan Kaufmann Publishers, 2000.
- [Ware, 2008]  
Ware, C., **Visual Thinking for design**, Morgan Kaufmann Publishers, 2008.
- [Van Wijk, 2005]  
Van Wijk J.J., **The Value of Visualization**. In: C. Silva, E. Groeller, H.Rushmeier (eds.), Proc. IEEE Visualization 2005, 2005.
- [Wilkinson, 1996]  
Wilkinson, L., **A graph algebra**. Computing Science and Statistics: Proceedings of the 28<sup>th</sup> Symposium on the Interface, 1996.
- [Wilkinson, 1999]  
Wilkinson, L., **The grammar of Graphics**. New York: Springer Verlag, 1999.
- [Young & Shneiderman, 1993]  
Young, D., Shneiderman, B., **A graphical filter/flow representation of Boolean queries: a prototype implementation and evaluation**. *J. Am. Soc. Inf. Sci.* 44, 6 1993.
- [Zuk, 2006]  
Zuk, T., Carpendale, S. T., **Theoretical Analysis of Uncertainty Visualizations**. In Proc. SPIE & IS&T Conf. Electronic Imaging, Vol. 6060: Visualization and Data Analysis 2006.

## Annexe



Visualisation polaire de trajectoires d'avions avec la transition vers la visualisation des types d'avion. FromDaDy

Dans cette annexe, nous avons utilisé et étudié le modèle de DataFlow pour la réalisation de visualisations. Le but de cette démarche est double. Premièrement nous pensons que si ce modèle permet la description d'une classe importante de visualisations, c'est qu'il peut être une base pour la caractérisation de visualisations.

Deuxièmement, son utilisation va nous permettre d'identifier des classes de visualisation et de comprendre plus précisément les spécificités de chacune d'entre elles.

# I. Validation avec le modèle de DataFlow

## A. Implémentation du modèle de DataFlow

De nombreux logiciels de visualisation, pour ne pas dire tous, utilisent le modèle de Data Flow pour la réalisation de visualisations de données. Nous avons choisi de développer notre propre logiciel afin de maîtriser chacune des étapes du modèle. Cette implémentation va utiliser un modèle simplifié de Data Flow avec une partie « données », une partie « paramétrages de la configuration visuelle » et une partie « filtrage des données ».

### 1. Les données

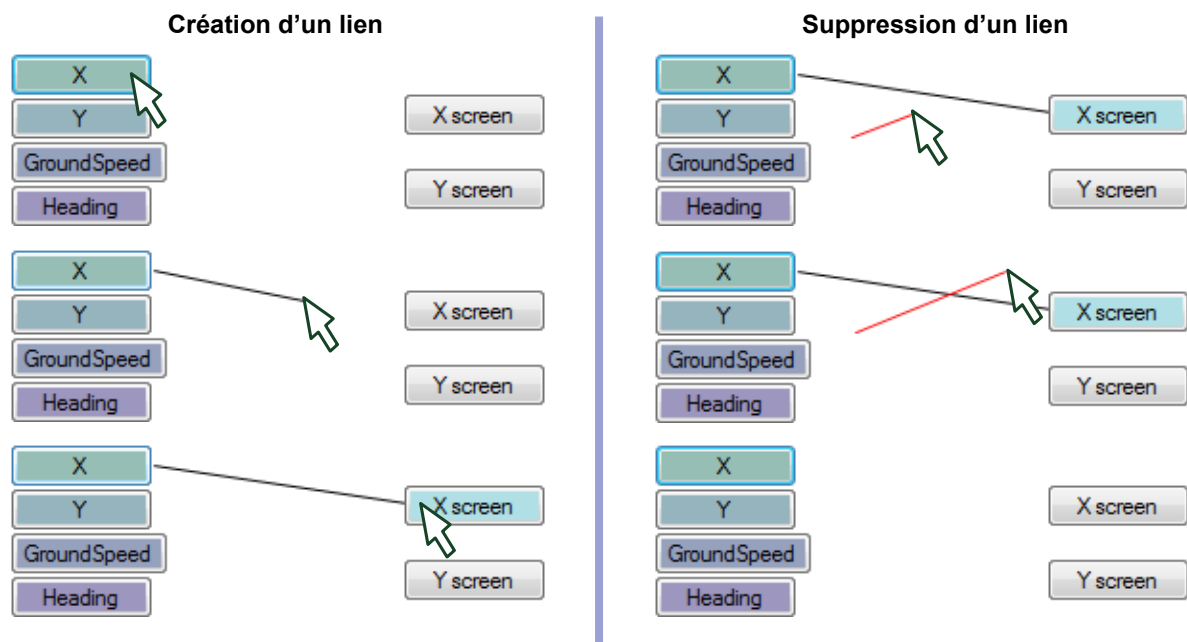
Ce logiciel prend en entrée tout type de données sous forme d'enregistrement. Le logiciel interprète les données de type tableur ou les fichiers au format XML.

Par exemple pour un fichier texte, chaque ligne correspond à un enregistrement. L'extrait suivant correspond à une série de 7 positions de l'avion AF1048 :

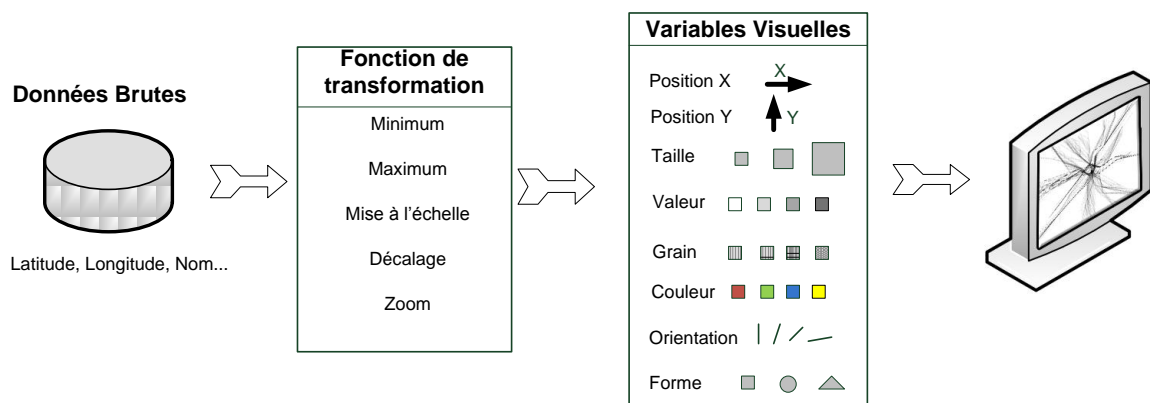
#vol	heure	Y	X	alti	indic	code
232	39436.21	64.11	-20.05	300	AF1048	7404
232	39494.43	59.56	-22.22	300	AF1048	7404
232	39521.19	57.28	-22.45	300	AF1048	7404
232	39617.27	49.55	-20.56	300	AF1048	7404
232	39686.94	45.28	-16.28	300	AF1048	7404
232	39793.35	43.84	-6.45	300	AF1048	7404
232	39793.92	43.77	-6.50	301	AF1048	7404

### 2. Les paramètres visuels

Le choix des paramètres visuels est basé sur les variables visuelles de Bertin [Bertin, 1983] et sur la théorie des échelles de Stevens [Stevens, 1946]. Dans notre logiciel, il est possible de faire des connexions (sous forme de lignes) entre des champs de données et des variables visuelles pour l'affichage. Ainsi, l'utilisateur tire un trait entre le champ des données X (Longitude de l'avion) et la variable visuelle X de l'écran pour afficher sur l'abscisse le champ X de toutes les données. Si l'utilisateur ajoute la connexion entre le champ nommé Y (Latitude de l'avion) et la variable visuelle Y de l'écran, l'image produite est un ensemble de points de coordonnées X, Y. Pour supprimer une connexion, l'utilisateur dessine un trait qui « coupe » une connexion.



Les connexions réalisées entre les champs de données brutes et les variables visuelles implémentent un modèle de Data Flow. Les données brutes sont ainsi affichées à l'écran à l'aide d'une fonction de transfert qui est, dans le cas de notre logiciel, implicite. Cette fonction fait correspondre la valeur minimale de la donnée brute à la valeur minimale de la variable visuelle. Il en est de même pour les marges supérieures. Par exemple, pour la connexion entre la Longitude et la variable visuelle X, le logiciel fait une mise à l'échelle de la longitude des données brutes de telle sorte que les longitudes utilisent toute la largeur de l'écran.



Pour compléter le choix des axes (variables visuelles de position X et Y), l'utilisateur peut aussi choisir des paramètres du design pour la réalisation de sa représentation des données. Il peut ainsi choisir la forme à afficher entre des points, des triangles, des rectangles, des chevrons, des lignes, ou du texte. En complément, il est possible de choisir la ou les couleurs d'affichage avec une correspondance entre les valeurs minimales et maximales d'un champ de données et deux couleurs en utilisant une fonction de transfert linéaire.

L'exemple suivant montre la fonction de transfert utilisée pour afficher une forme avec une taille qui varie en fonction de la vitesse de l'avion.

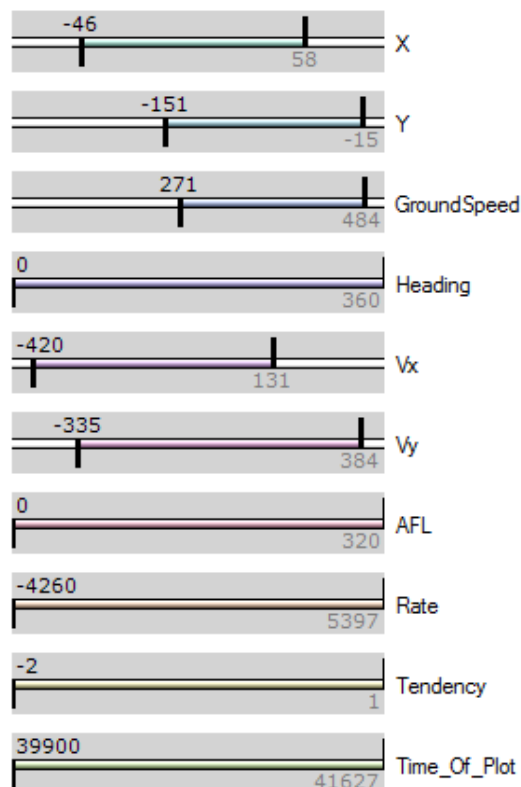


$$\text{Mise à l'échelle} \begin{cases} \text{Vitesse}_{\text{Min}} \rightarrow \text{Taille}_{\text{Min}} \\ \text{Vitesse}_{\text{Max}} \rightarrow \text{Taille}_{\text{Max}} \end{cases}$$

Des choix peuvent aussi être réalisés pour la taille des formes et leur orientation.

### 3. Le filtrage des données

A tout moment, l'utilisateur peut décider d'afficher tout ou une partie des données en procédant à des requêtes dynamiques sous forme de paramétrage de filtres sur les champs de données avec des *rangesliders*. Ainsi, si l'utilisateur souhaite n'afficher que les données qui ont une valeur de leur champ « GroundSpeed » (la vitesse des avions projetée au sol) entre 271 et 484 Kts, il modifie le début et la fin de la plage de sélection du *rangeslider* associé à ce champ.



## **B. Exemple de visualisation : Minard**

---

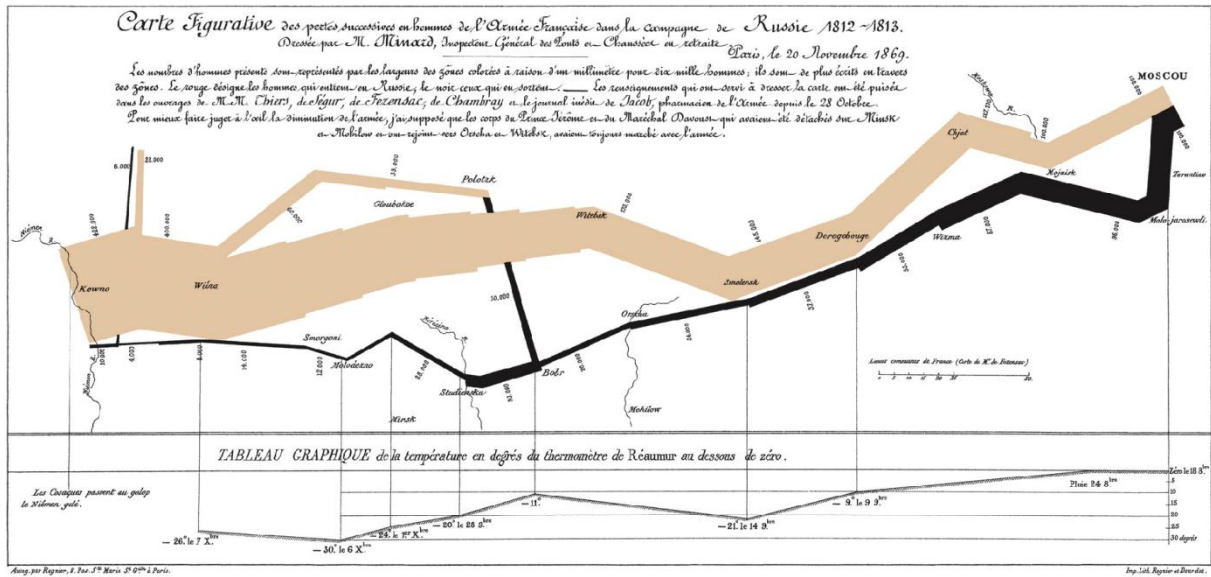
Comme première évaluation du modèle de Data Flow, nous avons décidé de réaliser l'illustration de la Marche de la Grande Armée produite par Minard. Cette image fait partie des visualisations de référence pour la transmission d'information.

### **1. Informations disponibles**

Charles Joseph Minard (1781-1870), ingénieur Français, a produit cette image qui retrace l'évolution de l'armée de Napoléon pendant sa campagne de Russie. Cette image a été réalisée en 1869 et illustre les pertes conséquentes en vies humaines en 1812 durant la tentative de conquête de la Russie par l'armée de Napoléon. Partant de la gauche de l'image avec la frontière Russo-polonaise à côté de la rivière Niemen, l'épaisseur du trait en ocre représente la taille de l'armée avec initialement, en 1812, ses 422 000 hommes. En septembre, l'armée, avec 100 000 hommes, atteint Moscou, ville déserte et dévastée. Le chemin de retour est dessiné avec une ligne de couleur noire reliée à une échelle supplémentaire en bas de l'image qui indique les températures à certaines dates. Cette année là, l'hiver a été glacial et beaucoup d'hommes sont morts de froid pendant le périple du retour. Comme l'indique le graphique, le passage de la rivière Berezina a été un désastre avec seulement 10 000 survivants. Cette image montre aussi les mouvements des troupes auxiliaires qui devaient assurer la protection des flancs et des avants de l'armée. L'image de Minard raconte une histoire riche et cohérente avec une utilisation judicieuse de paramètres graphiques. Pas moins de six informations sont représentées : l'effectif de l'armée, sa position (2 dimensions), la direction de l'armée, la température, la date.

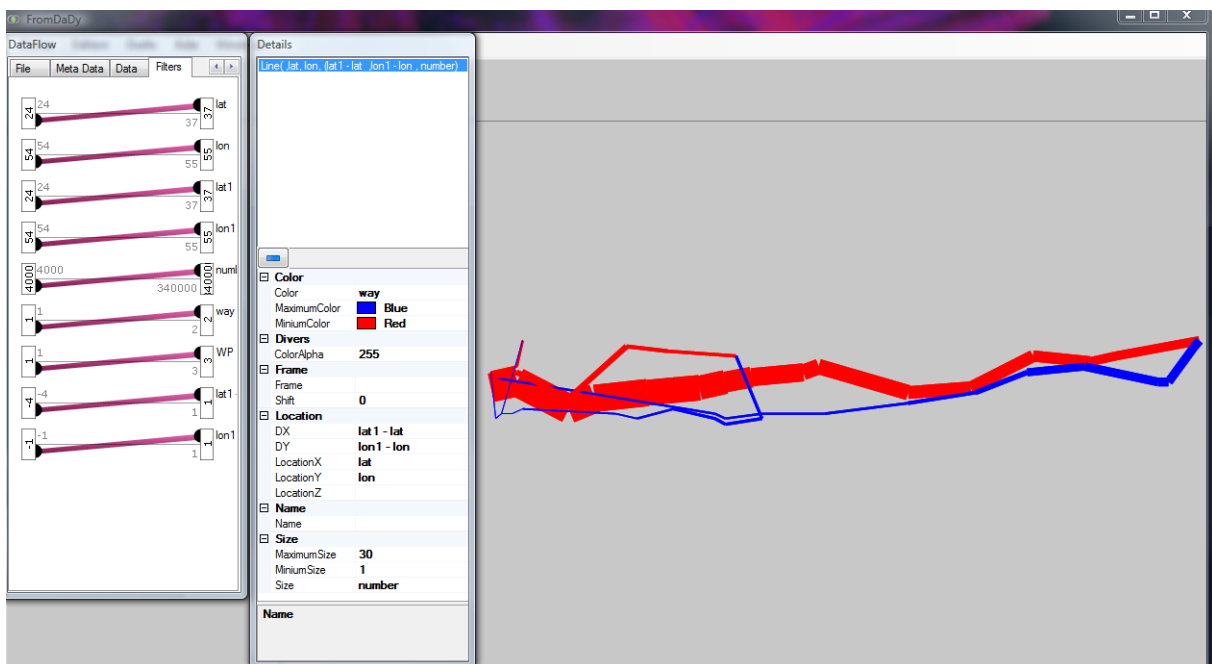
Cette représentation code des informations supplémentaires mais qui demandent une analyse de l'image :

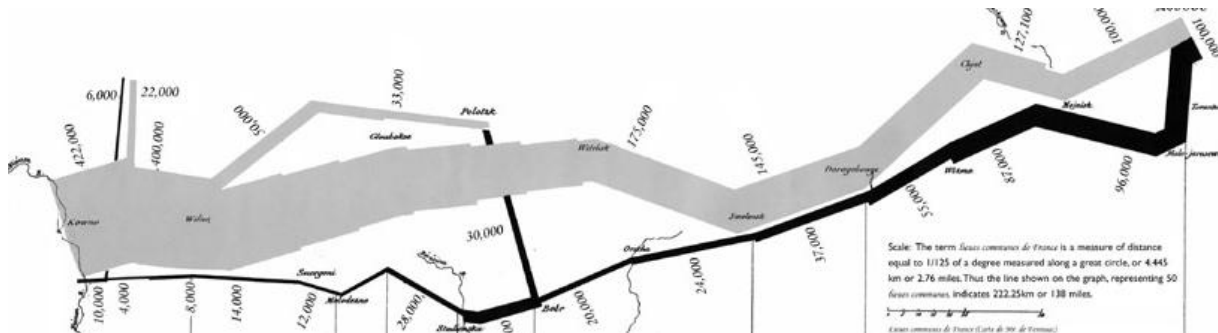
- En comparant les épaisseurs des traits de départ et d'arrivée de l'armée, on conclut rapidement que Napoléon n'a pas réussi à conquérir la Russie.
- La diminution continue de l'épaisseur du trait s'explique par la technique utilisée par l'Armée Russe : la terre brûlée. L'armée russe n'a pas combattu directement l'armée de Napoléon mais a fuit en brûlant tout sur son passage pour ne laisser aucune ressource disponible. L'armée de Napoléon a alors été progressivement décimée par la faim et la maladie.
- On observe une corrélation entre les passages de rivières et la diminution de la taille de l'armée. On observe aussi une corrélation entre les températures extrêmes et la diminution de la taille de l'armée.



Cette image a été réalisée avec les paramètres visuels suivants :

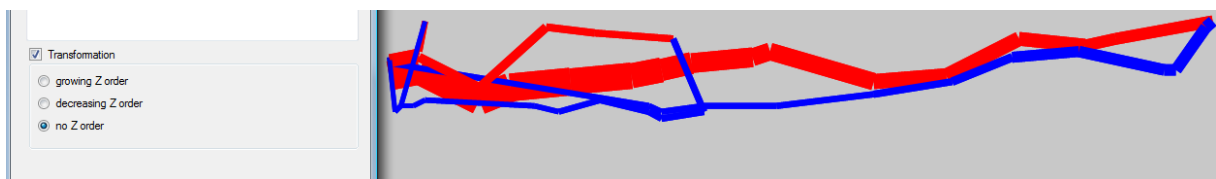
- Les positions de l'armée sont codées par une ligne (correspondance entre la Latitude-Longitude et le X-Y écran).
- La couleur code la direction de l'armée (rouge : attaque, bleu : retraite).
- L'épaisseur de la ligne code la taille de l'armée.



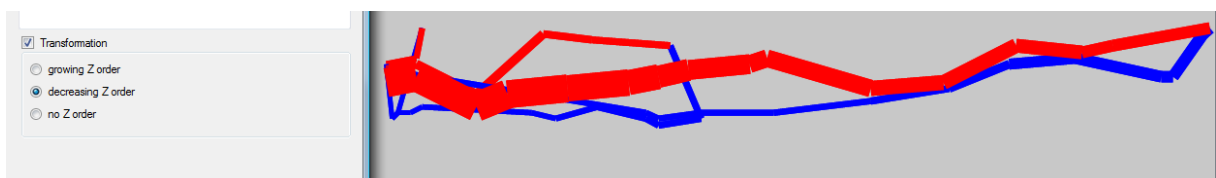


En comparant le dessin de Minard et la représentation des données avec un système de Data Flow, on remarque un certain nombre d'incohérences. Minard a choisi par exemple de ne pas entièrement respecter les données en les altérant pour des raisons de lisibilité du dessin : l'arrivée et le départ de l'armée à Moscou ne sont pas dessinés à partir du même point. Ce choix visuel peut être réalisé en modifiant les données à injecter au Data Flow. Une autre remarque concerne l'occlusion des traits. Le retour de l'armée (en noir) passe sous le chemin d'attaque. Il faut donc modifier notre système de Data Flow pour prendre en compte la spécificité de la classe de visualisation qui utilise l'occlusion.

Par défaut les données sont dessinées dans leur ordre d'arrivée dans le Data Flow :



Si on décide de les ordonner par rapport au champ direction (0 : attaque, 1 retraite) on obtient une image respectant l'occlusion choisie par Minard :



Il reste encore quelques erreurs visuelles dûes à notre méthode pour dessiner les lignes. Nous utilisons des sections de ligne qui ne se recouvrent pas totalement et laissent apparaître des discontinuités. Ces erreurs visuelles ne sont pas liées au DataFlow qui joue parfaitement son rôle et nous permet d'afficher correctement la représentation de Minard.

## C. Application du modèle de Data Flow aux visualisations de l'ATC

Généralement, les modèles de visualisation décrivent des systèmes dédiés à la recherche d'information ou à la production d'images. Les visualisations ATC sont différentes des systèmes de visualisation classiques sur plusieurs points.

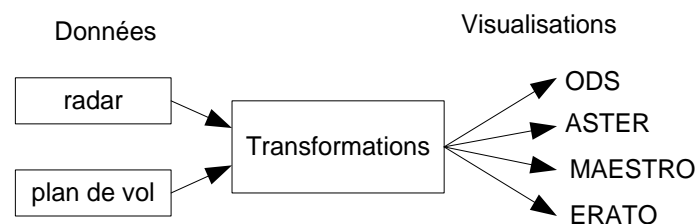
- elles sont dynamiques ; elles évoluent au cours du temps,

- elles sont considérées comme des systèmes de supervision car les transformations ne sont pas modifiées par l'utilisateur. L'utilisateur observe l'image mais ne la modifie pas,
- elles se ressemblent car elles se basent toutes sur les mêmes données, ce qui nous oblige à plus de précision pour différencier les caractéristiques de chaque représentation.

Les informations affichées proviennent soit des données radar, soit des plans de vol avions (succession de points de référence que doit suivre l'avion pendant son vol). Nous considérons que les données sont stockées dans une table, dont chaque ligne contient un enregistrement.

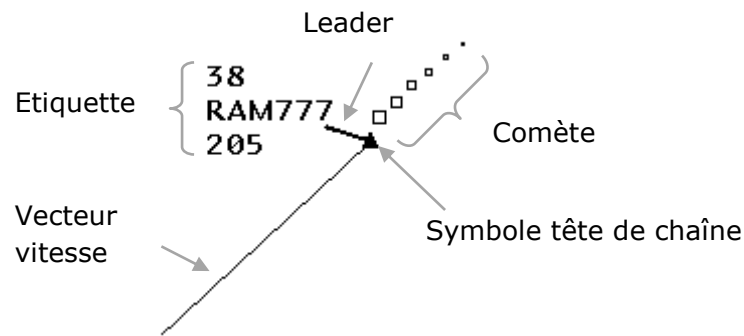
Données	Détails	Type
Code transpondeur	Identifiant avion	N
X	Longitude	Q
Y	Latitude	Q
Vitesse sol		Q
AFL	Niveau de vol (Altitude)	Q
Taux de montée	Vitesse verticale	Q
Temps	Heure du plot radar	O

Les données radar et plan de vol sont traitées pour être affichées. Ainsi sont produites, avec les mêmes données, les visualisations du domaine du Contrôle Aérien, comme par exemple des visualisations radar (ODS, ASTER), des séquenceurs au départ et à l'arrivée des aéroports (MAESTRO), des systèmes de prévisions et de gestion de conflits (ERATO).



## 1. Réalisation de l'image radar

Pour valider l'utilisation du modèle de Data Flow dans le domaine du Contrôle Aérien, nous allons réaliser la visualisation principale utilisée par les contrôleurs aériens : la vue radar qui affiche les positions courantes et passées des avions avec la piste radar. Ce terme est historique et ne fait pas référence à la zone de décollage et d'atterrissage des avions mais au système de suivi radar de la position des avions.



La position actuelle de l'avion est symbolisée par un triangle : le *symbole tête de chaîne*. Les positions passées sont représentées par des carrés de plus en plus petits en fonction de leur ancienneté. L'ensemble des positions passées forme la *comète radar*. Les informations de vitesse, de niveau de vol (Altitude) et l'identifiant de l'avion sont données sous forme textuelle et sont regroupées dans l'*étiquette*. Le *leader* connecte le symbole tête de chaîne à l'étiquette. Le *vecteur vitesse* part de la position de l'avion et s'arrête à sa future position dans 3, 6 ou 9 minutes si les paramètres de vol (cap et vitesse) ne sont pas modifiés.

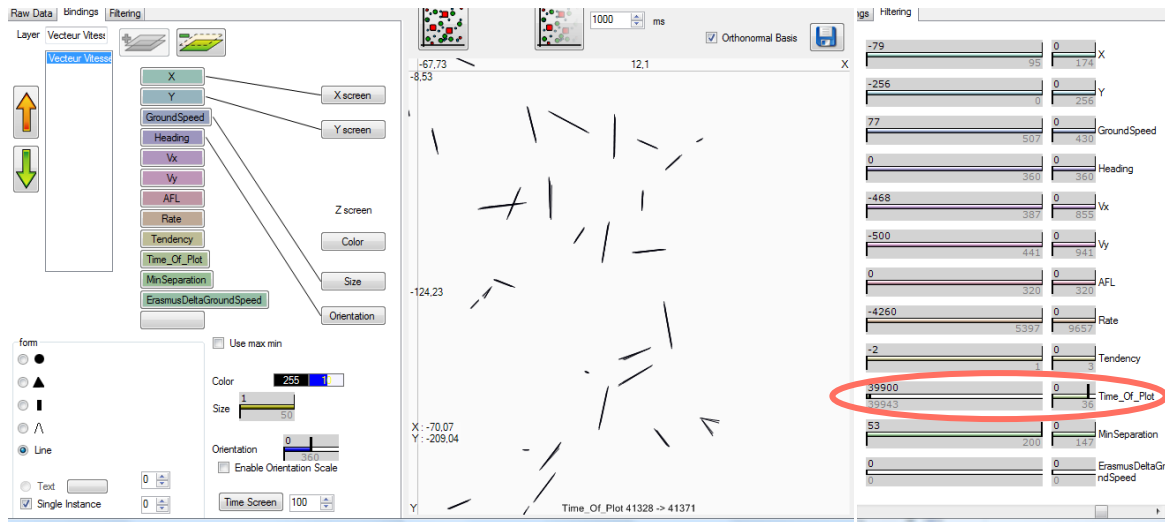
Pour réaliser cette image, nous avons dû réaliser trois fonctions de transformation des données vers un visuel. Pour ce faire, nous avons créé trois couches avec pour chacune un ensemble de choix de designs et des filtrages sur les données.

#### a) Réalisation du vecteur vitesse

Il existe plusieurs façons de réaliser le vecteur vitesse. Nous avons choisi d'utiliser les paramètres de position, taille, et orientation.

Le vecteur vitesse est représenté par une ligne (forme) positionnée en X-Y sur l'écran en fonction des champs X-Y de l'avion, avec une taille qui varie avec le champ GroundSpeed (vitesse de l'avion), et avec une orientation fonction du champ Heading (direction de l'avion).

Pour ne pas surcharger la vue, nous avons filtré les données pour n'afficher que les vecteurs vitesse à un instant (utilisation du filtrage dynamique sur le champ Time\_Of\_Plot). Ce filtrage sur les données n'est pas forcément optimal. Il implique que les données de tous les avions soient présentes à chaque instant. Si on veut afficher le vecteur vitesse de tous les avions à 12h50min10sec, il faut que les données de position, orientation et vitesse soient présentes pour tous les avions. Dans la pratique ce n'est pas le cas car les radars transmettent les données avions de façon asynchrone : pour obtenir les informations d'avions ayant volé à une heure précise, il faut effectuer un filtrage sur une plage d'heure et non sur l'heure. C'est pourquoi l'image suivante utilise deux rangesliders (heure + marge).

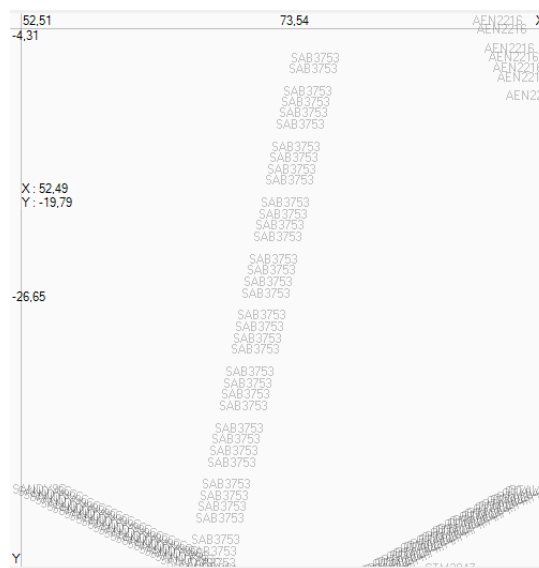


## 2. Réalisation de l'étiquette radar

L'étiquette radar regroupe les informations sur un avion. Elle est composée de la vitesse exprimée en Nœud sur la première ligne, du nom de l'avion avec son indicatif (compagnie aérienne avec un numéro de vol) sur la deuxième ligne, de son niveau de vol (l'altitude de l'avion) exprimé en pied sur la troisième ligne.

Pour simplifier la réalisation de l'étiquette, nous nous sommes cantonnés à afficher des informations textuelles : le nom de l'avion et son code transpondeur (code identifiant l'avion). Pour afficher l'étiquette, nous avons créé deux couches, une par information à afficher. Nous avons choisi d'afficher sur l'écran en X-Y les données des champs X-Y avec du texte correspondant aux champs SSR (code transpondeur) et CallSign (nom de l'avion).

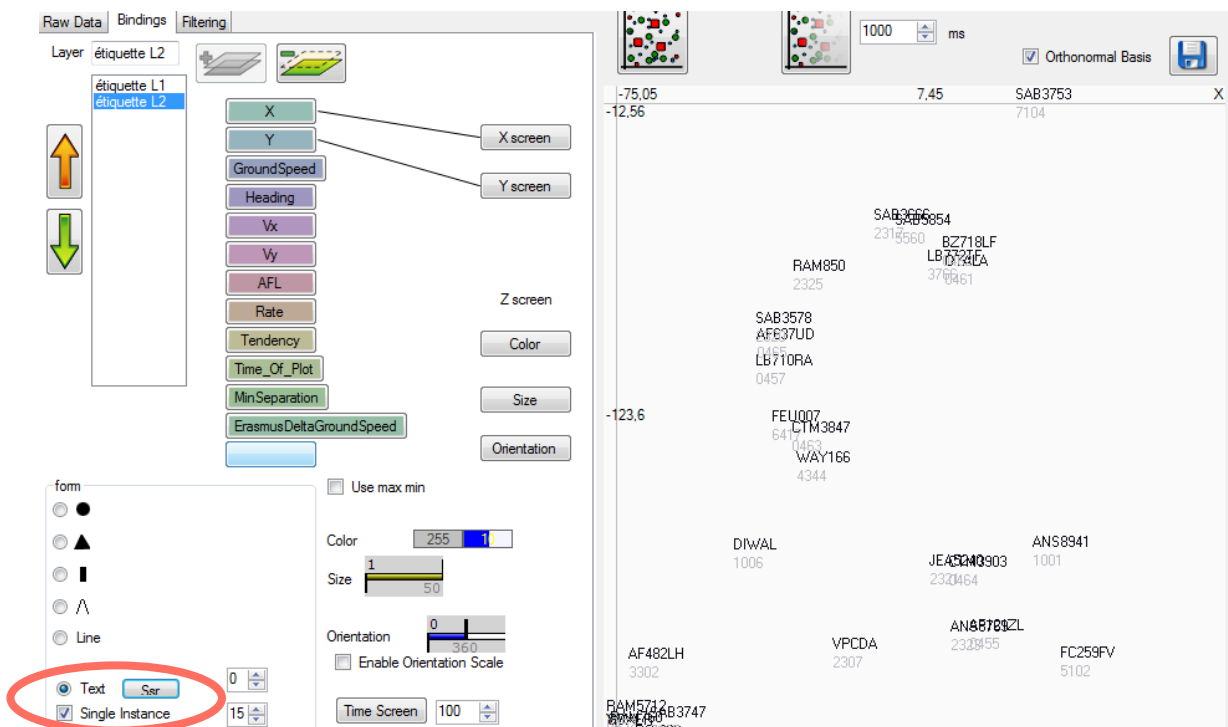
Nous retrouvons le même problème que pour le vecteur vitesse avec l'affichage de toutes les étiquettes, la visualisation est rendue inexploitable.



Pour y remédier, on peut utiliser un filtrage dynamique sur le temps, mais la visualisation n'est toujours pas optimale car les données ne sont pas uniformes : tous les champs des

enregistrements ne sont pas renseignés au même instant (time\_Of\_Plot), il en résulte des étiquettes sans information car renseignées plus tard.

La seule solution consiste à modifier le Data Flow en introduisant des systèmes de mémoire pour conserver des informations à ne pas dupliquer : dans notre cas, on affiche le CallSign de l'avion seulement s'il n'est pas déjà affiché. Une limitation de cette méthode est le choix de la position d'affichage de la donnée textuelle. Dans notre cas elle est affichée avec la première donnée contenant l'information textuelle de chaque avion. Notre jeu de données est ordonné dans le temps (car les données radar arrivent ordonnées dans le temps), il en résulte l'affichage du nom de l'avion sur le premier plot de chaque trajectoire. C'est un choix de design convenable mais non paramétrable avec notre modèle de mémorisation.

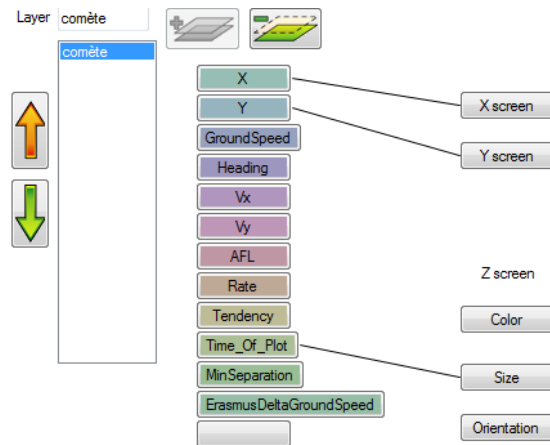


### 3. Réalisation de la comète radar

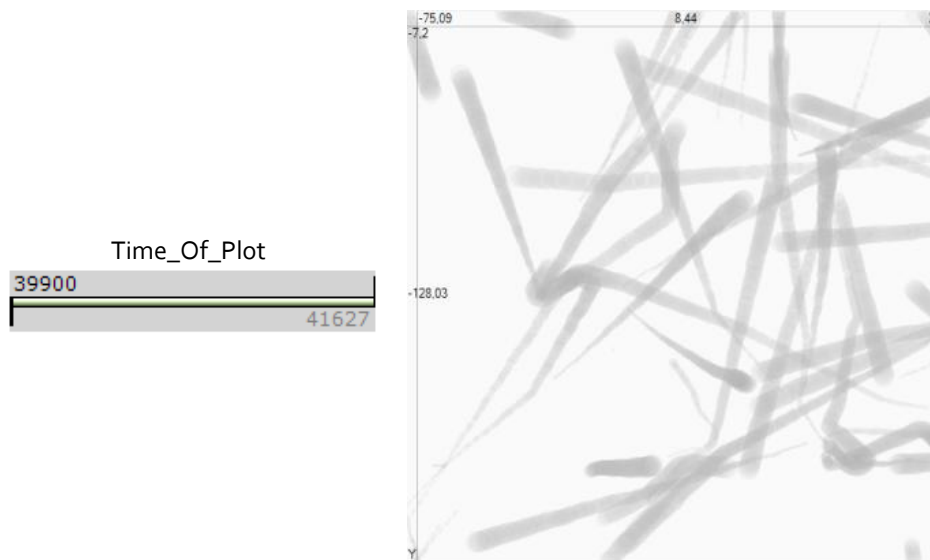
La réalisation de la comète est un peu plus complexe car la fonction de transformation fait intervenir le temps courant. En effet, la comète est composée de plusieurs points qui correspondent aux positions passées de l'avion. Chaque position a une taille qui dépend de l'ancienneté du plot radar (par rapport au temps courant) : la plus grande taille correspond à la position la plus récente, la plus petite, à la position la plus ancienne.

Les connexions entre les champs des données et les variables visuelles sont simples :





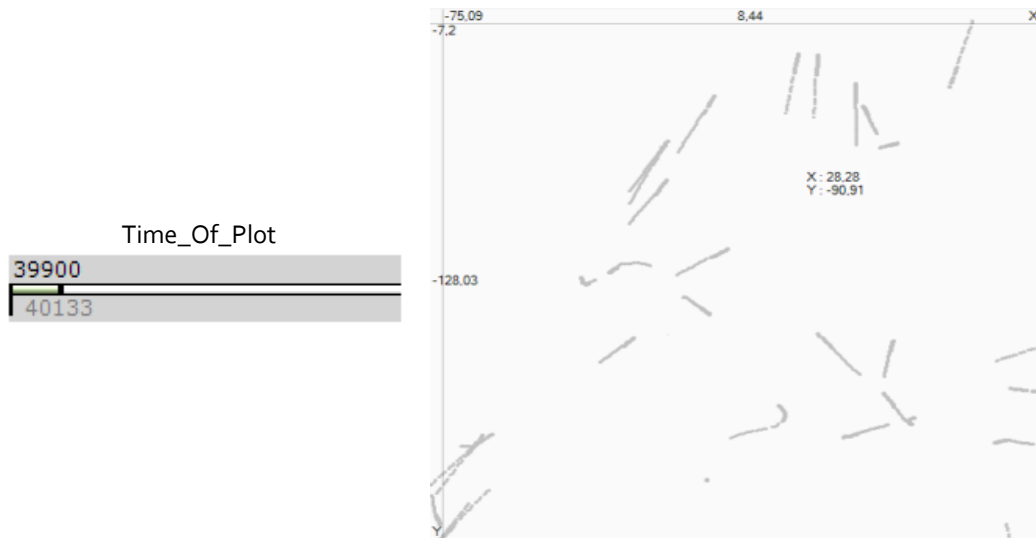
L'image suivante correspond à l'affichage d'une comète par avion avec toutes les données du jeu (dans ce cas une heure d'enregistrement radar au-dessus de Bordeaux).



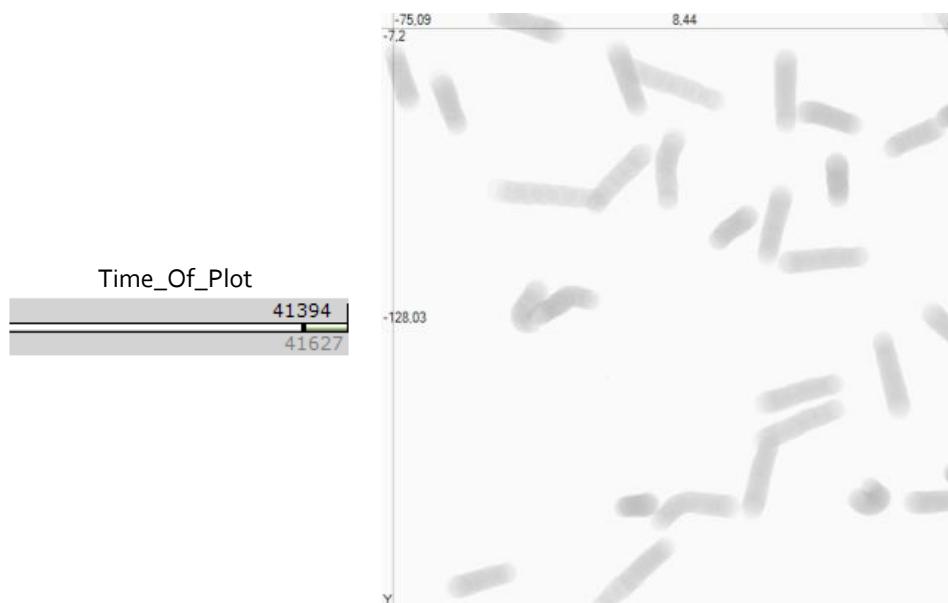
Pour la création de la comète, nous avons besoin des positions des avions entre deux instants correspondant au début et la fin de la comète. Ensuite, il faut que la taille de chaque plot corresponde à son ancienneté. Nous sommes obligés d'utiliser le filtrage pour n'afficher que les plots radar concernant la comète entre deux instants : filtrage de l'heure du plot radar avec comme borne supérieure l'heure du plot radar pour la tête de la comète et comme borne inférieure l'heure du plot radar de la queue de la comète.

De plus, la taille de chaque point doit être calculée en fonction du champ *Time\_Of\_Plot* en prenant garde d'obtenir la taille maximale et minimale en fonction des bornes du filtre. Les deux images suivantes correspondent à l'affichage de la comète entre les bornes définies par le rangeslider *Time\_Of\_Plot*. Dans ce cas la taille des plots radar est directement proportionnelle aux valeurs des bornes du rangeslider, alors que le visuel recherché doit produire des plots avec une taille fixe : les plots radar qui ont le champ *Time\_Of\_Plot* égal à la valeur minimale du rangeslider doivent avoir la taille minimale à l'écran, les plots radar qui ont le champ *Time\_Of\_Plot* égal à la valeur maximale du rangeslider doivent avoir la taille maximale à l'écran.

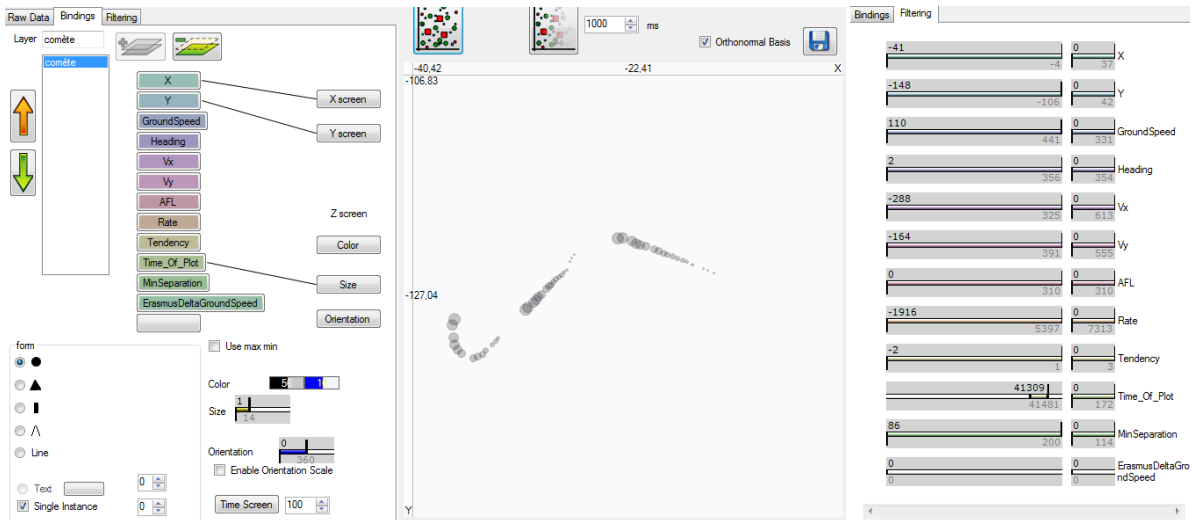
L'image suivante illustre le filtrage avec une sélection sur la plage basse des heures. Il en résulte une faible taille de comète sur toute sa longueur.



L'image suivante illustre le filtrage avec une sélection sur la plage haute des heures. Il en résulte une taille de comète épaisse sur toute sa longueur.



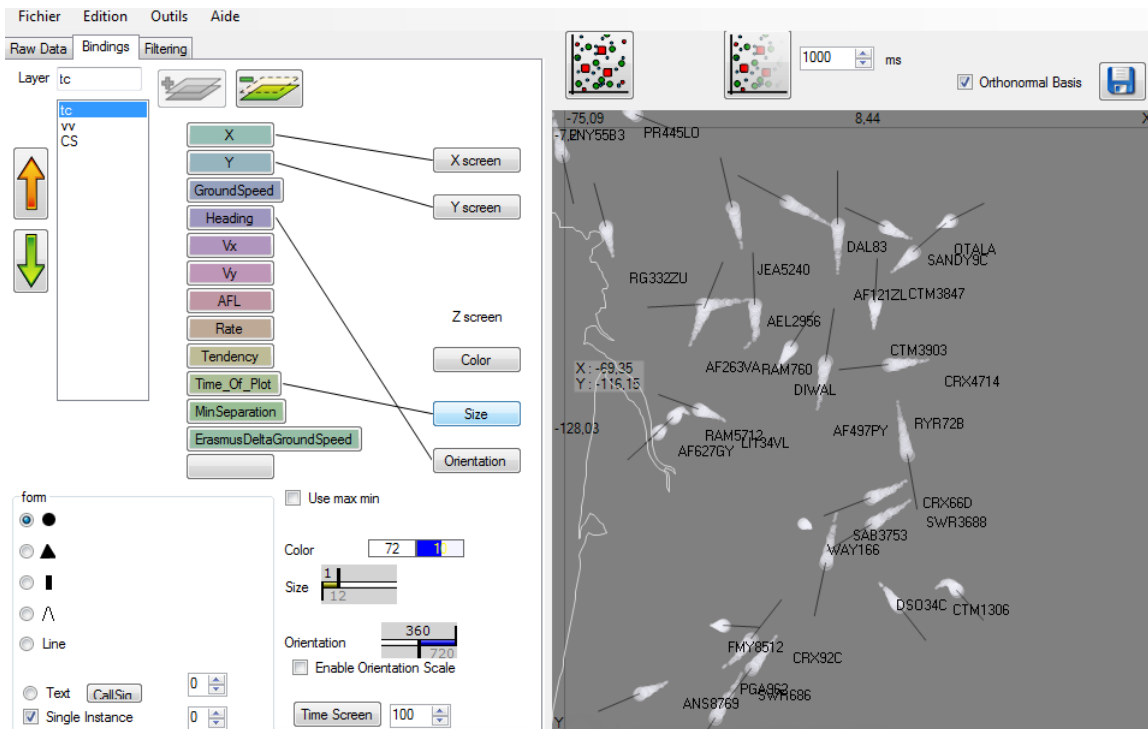
Si on prend en compte les valeurs des bornes de sélection du rangeslider Time\_Of\_Plot, on obtient une image qui produit la comète avec les bons paramètres de design.



#### 4. Image radar complète

Un fois toutes les couches assemblées, nous obtenons l'image radar utilisée par les contrôleurs aériens :

- Une couche pour le vecteur vitesse,
- Une couche pour la comète,
- Une couche pour l'étiquette radar.

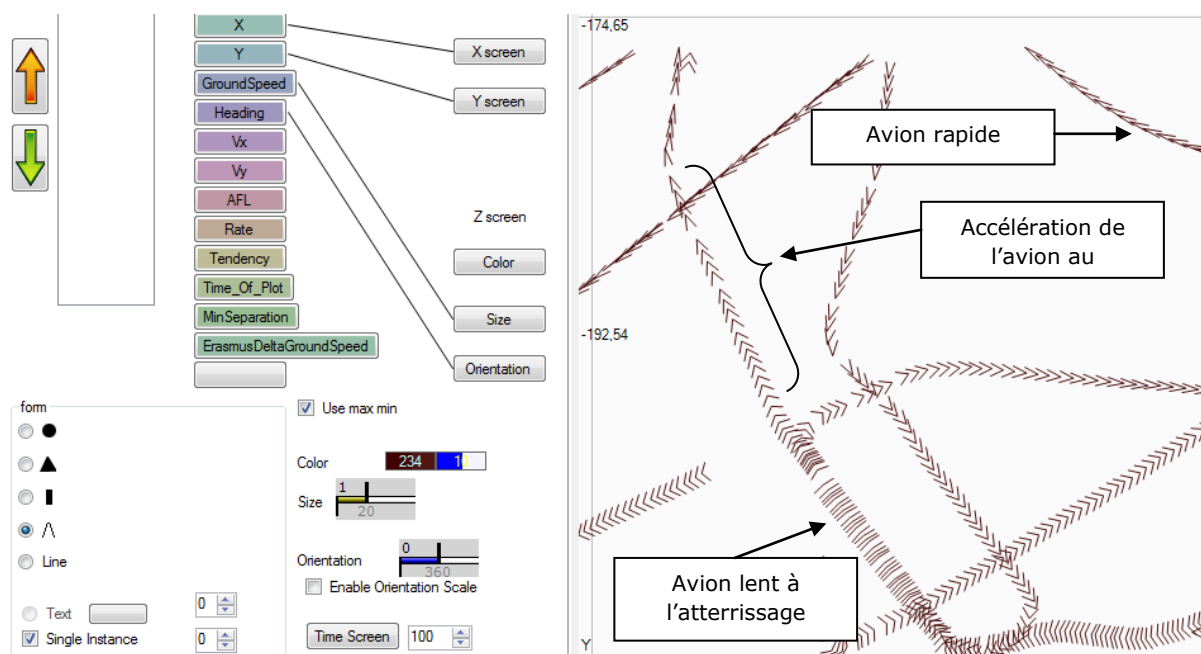


## D. Conception de nouveaux designs

Les exemples précédents illustrent l'utilisation du modèle de DataFlow pour reproduire des visuels existants, mais par extension, il est possible de réaliser des visualisations prospectives, c'est-à-dire des visualisations qui n'existent pas dans le domaine opérationnel du Contrôle Aérien.

### 1. Les chevrons radar

Nous avons développé un nouveau design pour représenter la position de l'avion avec un chevron. Ce dernier permet d'afficher la direction de l'avion (orientation du chevron), sa vitesse (avec le degré d'ouverture du chevron). Ce design a des propriétés dites émergentes car il permet de coder visuellement les variations de vitesse et donc d'en déduire les phases de vols de l'avion (décollage, atterrissage) et même dans certains cas la catégorie. Ce design est écologique car il utilise le phénomène physique de la résistance à l'air d'un chevron pour la représentation visuelle de la vitesse de l'avion : les chevrons très pointus offrent très peu de résistance à l'air, ils vont donc très vite, les chevrons très aplatis offrent une grande résistance à l'air et vont très lentement. L'ouverture et la fermeture du chevron sont contrôlées par la variable visuelle de taille (qui ne correspond plus à la définition de Bertin mais que nous avons conservée pour simplifier l'interface utilisateur).



### 2. Le disque des vitesses

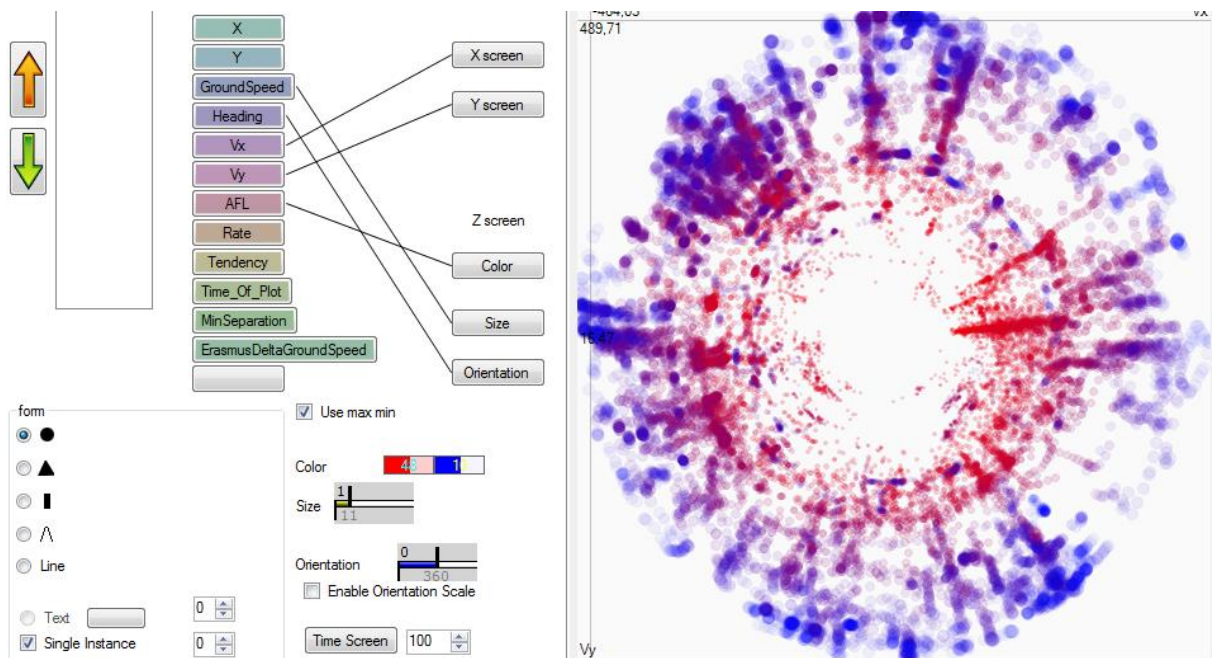
Notre logiciel permet d'afficher n'importe quelle donnée brute avec des paramètres de design issus des Variables Visuelles de Bertin. Au cours de la manipulation des paramètres de design, nous avons trouvé une représentation remarquable d'un point de vue visuel car elle trace un disque. Nous l'avons réalisé à l'aide des données  $V_x$  et  $V_y$  qui sont les composantes projetées de la vitesse sur l'axe des abscisses et des ordonnées. L'obtention d'un disque est compréhensible car le vecteur vitesse peut se calculer avec la direction de l'avion et ses vitesses projetées :

$$V = V_x \cos(\text{angle}) + V_y \sin(\text{angle})$$

Chaque point du disque est donc positionné en fonction de la direction de l'avion. Les avions volent dans toutes les directions, c'est pourquoi on obtient un disque. Les zones de fortes densités qui correspondent aux directions privilégiées des vols (les orientations des pistes d'atterrissage et de décollage par exemple).

Nous avons ajouté dans l'image suivante l'altitude de l'avion codée avec la couleur (les avions qui volent bas sont en rouge, et ceux qui volent haut sont en bleu) et la taille des points radar qui nous informe de la vitesse (faible vitesse avec une taille réduite et forte vitesse avec un diamètre de point important).

On constate que les points bleus (les avions qui volent haut) sont larges (ils volent vite). Ceci s'explique car les performances des avions sont meilleures en altitude (moins de prestances au frottement). Cette vue peut ainsi être exploitée pour trouver des avions qui ne respectent pas les contraintes environnementales : par exemple des avions rouges avec une taille de point importante.

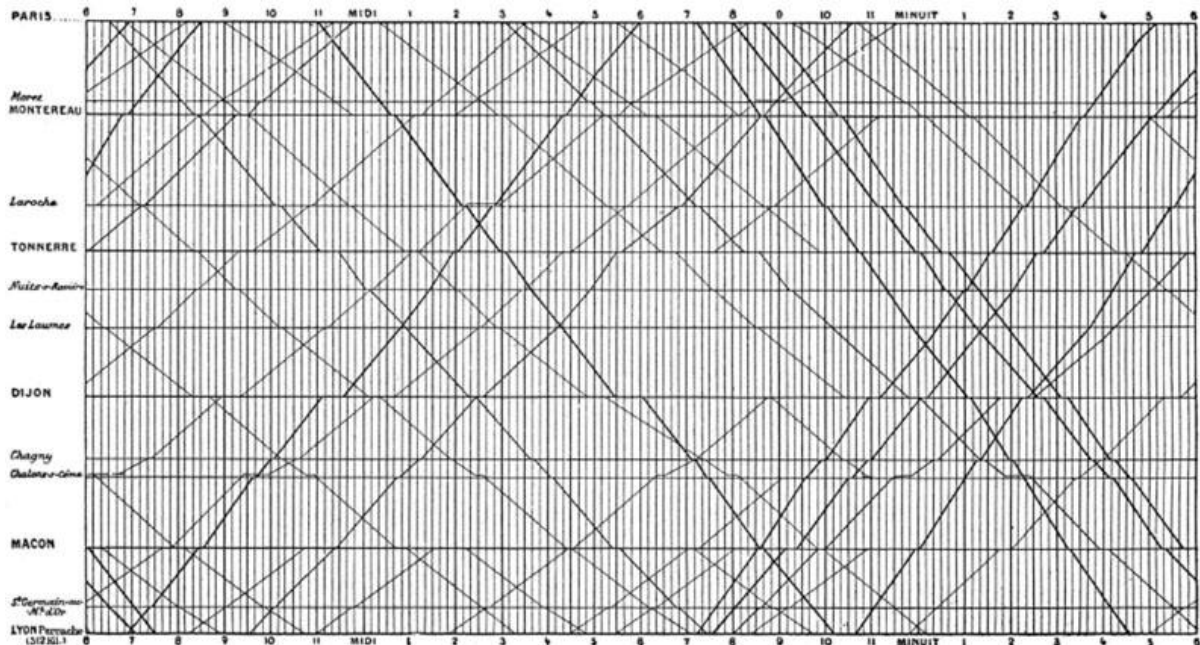


### 3. L'occupation spatiale des voies aériennes

Cette visualisation s'inspire de la représentation créée en 1880 par Etienne-Jules Marey sur les horaires des trains entre Paris et Lyon [Tufté, 1983]. C'est un graphe à deux dimensions qui place chaque ligne de train selon sa distance des gares de Lyon et Paris sur l'axe de Y et selon son heure sur l'axe des X. Le parcours de l'axe des Y donne toutes les gares disponibles sur le trajet Paris-Lyon. Le parcours de l'axe des X donne toutes les heures d'une journée. L'orientation des lignes permet de connaître la direction des trains : les lignes qui partent du haut à gauche vers le bas à droite correspondent aux trains en direction de Lyon, les lignes qui partent du bas à gauche vers le haut à droite correspondent aux trains en direction de Paris. De plus, les lignes horizontales indiquent que le train s'est arrêté à une gare donnée et la longueur de cette ligne horizontale indique le temps d'arrêt. La pente des lignes indique la vitesse des trains avec des arrêts

si elles comportent des sections horizontales. Des informations plus globales sont aussi fournies sur la fréquence des trains sur une journée grâce à la densité des lignes.

L'image suivante est la représentation originale de Marey. Tuftes en a critiqué la couleur de fond pour les gradations du graphe en lui préférant une autre couleur que celle utilisée pour dessiner les lignes de train, comme par exemple un marron clair en application de ses principes de qualité graphique.

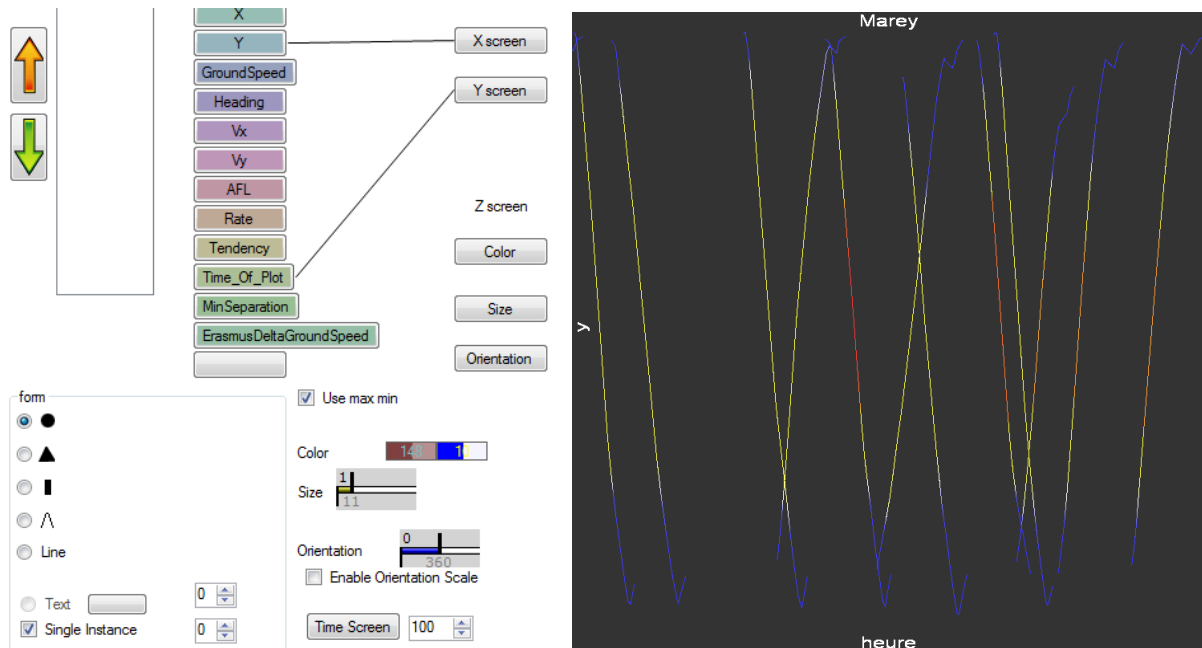


L'application au domaine de l'ATC n'est pas encore évaluée mais cette classe de visualisation peut aider à mettre en avant les charges de trafic sur les routes aériennes et ainsi appuyer les décisions concernant la répartition des avions sur une journée.

Dans l'image suivante, la couleur indique l'altitude de l'avion avec :

- du bleu pour les altitudes basses (l'avion est au sol à son aéroport de départ ou d'arrivée),
- du rouge pour les altitudes hautes (l'avion est à un niveau de vol élevé, il est en croisière).

Dans cette image, on voit apparaître clairement que certains avions volent plus haut que d'autres, certains sont plus rapides. On voit apparaître le sens de déplacement les fortes densités de rotation d'avions mais aussi les séquençements des départs et arrivées (souvent un avion part de la région lyonnaise suivie par un avion qui arrive de la région parisienne). Cette image permet d'analyser finement les séquençements réels des avions sur une voie aérienne donnée.



Même si ce n'est pas le propos dans ce chapitre, il est intéressant d'identifier les informations exploitables dans ce type de visualisation. On peut, par exemple, voir que le matin il n'y a que des vols au départ de Paris, ensuite on retrouve le hub de midi et le soir plusieurs avions partent quasiment en même temps. Sur la base de ces informations, un gestionnaire pourrait regrouper ces vols avec des appareils plus gros.

## E. Conclusion

Notre démarche a été d'implémenter le modèle de Data Flow pour valider, avec notre expérience, les contraintes liées à son utilisation. Cette approche nous a permis d'explorer plusieurs domaines : le design avec l'utilisation des variables visuelles, la caractérisation des visualisations, l'utilisation du modèle de Data Flow, la qualité des données, l'exploration des données...

### 1. Le design

Notre logiciel permet l'utilisation de variables visuelles pour représenter n'importe quel champ de données. Cette connexion est rendue explicite avec la création de liens par l'utilisateur. Nous avons généralisé l'utilisation des variables visuelles et ainsi permis à l'utilisateur d'essayer plusieurs designs. Cette utilisation du logiciel permet le prototypage de visuels avec des caractéristiques de design explicites et paramétrables.

### 2. La caractérisation

Le paramétrage des choix de design est en soi une forme de caractérisation ; cette description est nécessaire et suffisante pour la production d'une visualisation. Elle regroupe alors l'ensemble des informations qui compose la visualisation ainsi que le détail de son design.

Cette caractérisation est critiquable sur plusieurs points ; elle n'est ni formatée, ni synthétique. Son exploitation en est alors difficile. La comparaison de deux

caractérisations n'est pas directement réalisable. De plus, rien ne prouve que toutes les informations que contient cette caractérisation soit exhaustive.

Néanmoins, cette première caractérisation permet de contenir des informations remarquables pour chaque design et ainsi permet leur différenciation.

### **3. L'utilisation du modèle de DataFlow**

Nous avons recréé à l'aide du modèle de Data Flow l'image radar utilisée par les contrôleurs aériens et la marche de Napoléon. Pour ce faire, nous avons dû apporter un certain nombre de modifications au modèle :

- utilisation d'un élément mémoire conservant le nom de l'avion pour ne l'afficher qu'une seule fois dans l'étiquette,
- utilisation du filtrage dynamique et usage spécifique des paramètres de filtrage pour afficher la taille correcte de la comète radar,
- utilisation d'une variable de profondeur pour ordonner les dessins des formes à l'écran (Minard),
- utilisation de couches contenant la description d'une entité visuelle,
- utilisation des techniques d'animation.

Le modèle de Data Flow semble donc adapté pour la production de visualisations. Nous avons seulement mis en évidence quelques problèmes déjà étudiés et traités par les toolkits de visualisation [Baudel, 2002] [Fekete, 2004] [Wilkinson, 1999].

### **4. Les données**

L'utilisation du modèle de DataFlow a aussi mis en évidence les problématiques liées aux données et à leur qualité. Pour produire l'affichage des comètes radar, les données doivent être échantillonnées avec un taux constant mais aussi interpolées. L'ordre de traitement dans le modèle du DataFlow n'est pas sans conséquence sur la visualisation produite.

Pour afficher la comète radar, nous avons utilisé des zones mémoires. Cette astuce pose un problème lors de la navigation dans les données. Le sens de parcours modifie la visualisation. Cette problématique est à prendre en considération pour garder la cohérence entre le visuel et la base de données.



## **Caractérisation de visualisations et exploration interactive de grandes quantités de données multidimensionnelles**

Un grand nombre d'activités utilise des outils informatisés s'appuyant sur des représentations graphiques interactives. En tant que vecteurs de transmission d'informations, ces visualisations visent à optimiser la bande passante entre le visuel et l'utilisateur. Pour créer des visualisations efficaces, les concepteurs utilisent des méthodes basées sur leur expertise et sur des tests utilisateurs. Ces méthodes altèrent le processus de design : elles sont chronophages et ne reposent pas sur des fondements théoriques précis qui pourraient éviter des erreurs de conception en amont. Afin de mieux étudier les visualisations, nous proposons de les caractériser, c'est-à-dire d'en énumérer et d'en définir systématiquement les propriétés, tout en explicitant les phénomènes visuels qui engendrent la transmission d'informations. La thèse que nous soutenons consiste à dire que la caractérisation des visualisations est importante pour leurs conceptions, et que son opérationnalisation par des méthodes et des outils permet de concevoir de meilleurs systèmes interactifs.

Ainsi, nous avons déterminé qu'une caractérisation est composée de deux éléments : son contenu et sa mise en forme. Nous montrons que le modèle de Data Flow, modèle usuel permettant la description de la construction de visualisations, ne permet pas de les caractériser. Nous proposons d'autres modèles qui permettent de caractériser l'aspect dynamique, les informations émergentes (et notamment celles issues des visualisations dites "écologiques"), ainsi que l'accumulation graphique. En utilisant des visualisations issues du milieu du transport aérien, nous montrons comment la caractérisation permet de comprendre leurs similitudes et leurs différences, de lever les ambiguïtés de communication entre concepteurs, et de concevoir, par exemple, une nouvelle visualisation schématique de routes aériennes. Enfin, nous avons réifié le processus de transformation d'un ensemble de données vers une visualisation sous la forme d'un logiciel d'exploration et de visualisation de grandes quantités de données multidimensionnelles : From Data To Display (FromDaDy). Ce logiciel s'appuie notamment sur l'exploration des configurations visuelles, et sur un ensemble d'interactions originales avec une implémentation performante. FromDady permet de configurer et de manipuler en temps réel des visualisations riches et originales, comme celles utilisant l'accumulation.

**Mots clés** : Interaction Humain-Machine (IHM), Visualisation d'Informations (InfoVis), Design, Caractérisation, Exploration de données, Contrôle Aérien.

## **Characterization of Visualizations and Interactive Exploration of Large Quantity of Multidimensional Data**

A great number of activities use IS (Information Systems) relying on interactive graphic visualizations. As a vector for the transmission of information, these visualizations aim at optimizing the bandwidth between the user and the visual information. To create effective visualizations, designers use methods based on their expertise and user tests. These methods affect the design process, they are time consuming, and they do not rely on specific detailed theories, which could avoid, in advance, errors of conception. In order to better study these visualizations, we propose to characterize them, which means to systematically number and define their properties explaining all the visual phenomena that enable transmission of information. The thesis that we hold to, consists in saying that the characterization of a visualization is important for their conception, and that their exploitation in the light of these methods and tools enable the creation of better interactive systems.

Thus, we have determined that a characterization is composed of two elements: its contents and the way in which its content take form. We will show that the DataFlow model, a model that usually permits the description of the construction of the visualization, does not accomplish a sufficiently complete characterization. We will propose other models enabling characterization of the dynamic aspects, emerging information (notably those issuing from so-called "ecological" design) and especially graphical accumulation. In using visualization issuing from the milieu of air transportation, we will show how the characterization enables the understanding of their similarities and differences, to remove the ambiguities from the communication between designers and to conceive of, for example, a new schematic visualization of air routes. In addition, we have adapted the process to transform a collection of data into a visualization in the form of an exploration software of large numbers of multidimensional data. This software relies notably on an exploration of visual configuration and on a collection of original interactions with powerful implementation. FromDaDy (From Data to Display) enables the configuration and manipulation, in real-time, of rich and original visualizations, such as those using graphical accumulation.

**Keywords** : Human-Computer Interaction (HCI), Information Visualization (InfoVis), design, characterization, data exploration, Air Traffic Control (ATC).