Hervé GOËAU

MASTER 2 Pro Audiovisuel & Multimedia – TvNi
DREAM – Le Mont Houy
Université de Valenciennes et du Hainaut Cambrésis
BP05. 59313 VALENCIENNES

VISUALISATION SPATIO-TEMPORELLE DE SEQUENCES AUDIOVISUELLES

Institut National de l'Audiovisuel
Direction Recherche et Expérimentation
4 avenue de l'Europe
94366 Bry-sur-Marne

Stage de fin d'études effectué du 9 février au 23 juillet 2004
Responsables : Agnès Saulnier, Olivier Buisson
Enseignant responsable : François-Xavier Coudoux

REMERCIEMENTS

Tout d'abord un grand merci à mes tuteurs qui m'ont permis de travailler sur un sujet de stage de fin d'études passionnant alliant technicité et culture audiovisuelle : Olivier Buisson pour sa rigueur scientifique, Agnès Saulnier et Marie-Luce Viaud pour leurs soutient et encouragements. J'ai particulièrement apprécié leur vision globale de l'audiovisuel et intuition sur des outils de représentation de scène vidéo.

Je remercie également M.Coudoux pour son attention et son investissement en tant que responsable pédagogique dans cette nouvelle formation de Valenciennes.

Un remerciement enfin à toute l'équipe, chercheurs, doctorants, et stagiaires pour avoir contribué à un cadre de travail agréable, Alexis Joly, Jean-Etienne Noiré et Jérôme Thièvre, pour l'aide qu'ils ont pu m'apporter et leur disponibilité.

SOMMAIRE

RESUME

La profusion de documents audiovisuels produits et diffusés quotidiennement pose d'importantes problématiques de recherche et de consultation des œuvres après leur archivage. De manière analogue au domaine textuel, des expérimentations et études sont en cours dans la communauté scientifique et industrielle, autour du MPEG-7 par exemple, pour fédérer un mode d'accès au contenu audiovisuel. En parallèle de cette démarche, des outils de visualisation sont étudiés et développés afin de permettre une consultation rapide de vidéo. Ce domaine de recherche propose une réflexion sur la représentation condensée d'une séquence animée en créant des « résumés » vidéo.

Historiquement, l'image « mosaïque », c'est-à-dire l'extraction et la juxtaposition d'images clés d'une vidéo, constitue un premier objet audiovisuel allant dans ce sens, explorant ainsi la composante temporelle. Plus récemment sont apparus des systèmes de résumé vidéo dynamique ainsi que quelques tentatives de synthèse d'images descriptives. Par ailleurs, la construction de panoramas et la fusion d'images, notamment étudiées en vision robotique, exploitent la composante spatiale d'une vidéo pour mettre à plat une scène filmée et apparaît comme une façon intéressante de réduire les scènes audiovisuelles.

L'objectif de ce stage est de proposer des modes de visualisation en exploitant ce principe de mise à plat de vidéos dans le domaine de l'audiovisuel. En effet, la diversité et la richesse de plans qu'induisent les différents registres de réalisations audiovisuelles soulèvent de nouveaux enjeux scientifiques sur la fusion d'images.
Pour finir, c'est sur fond de développement d'un mini-logiciel, qu'un début de réflexion sera menée sur l'analyse, l'interprétation et l'utilisation de ces images « mosaïques étendues».

MOTS CLES

Images mosaïques – skimming – panoramas – signature - estimation de mouvements - homographies et transformées géométriques d'images – appariement d'images – détections de points d'intérêt – algorithme RANSAC – « motion mosaïc » - sommaire vidéo – mise à plat – images descriptives

PLAN DU DOCUMENT

Après une brève présentation de l'entreprise et du département Recherche, nous décrirons les deux équipes d'accueil et leurs activités.

Dans la deuxième partie, nous étudierons la technique de signature d'image développée à l'INA.

La troisième partie propose un état de l'art recensant différentes techniques pertinente pour la construction d'image spatio-temporelles à partir de séquence vidéo et discute de l'utilisation des techniques de signature d'images dans ce contexte.

La quatrième partie décrit la mise en œuvre d'un environnement développé en langage C afin de tester des algorithmes inspirés de l'état de l'art.

La dernière partie s'appui sur les résultats obtenus pour extraire les problématiques inhérentes de ces objets audiovisuels, tant au niveau technologique qu'au niveau utilisation et interprétation, afin d'orienter au mieux la poursuite du projet.

Enfin ce rapport de stage se termine par plusieurs annexes contenant notamment les principales fonctions implémentées.

INTRODUCTION

L'engouement actuel pour les appareils numériques audio-vidéo, la profusion de chaînes de télévision et la production cinématographique engendrent à l'échelle mondial un volume horaire vertigineux alimentant indéfiniment de nombreuses banques de données.

Les organismes de sauvegarde du patrimoine audiovisuel, tel que l'INA, se trouvent alors confrontés à d'importantes problématiques liées à la recherche, à l'accès et à la consultation de vidéos parmis les millions d'heures archivées.

Il est donc naturel pour l'INA, possédant l'une des plus grande banque audiovisuelle au monde, d'axer les thèmes de son département Recherche sur l'indexation, la restauration et la visualisation de films et vidéos.

L'outil informatique permet d'envisager et de créer une alternative pour la recherche de document. Pour l'heure, les applications les plus flagrantes sont les moteurs de recherche sur le web. D'autres équipes travaillent sur la nouvelle norme MPEG7 qui permettra une recherche non plus textuelle, mais de plus haut niveau, sur la reconnaissance de formes visuelles et sur les empreintes sonores.

Conjointement à cet engouement pour l'indexation et l'analyse d'images, le thème du « résumé vidéo » propose des outils de visualisation permettant de modifier le processus de consultation d'une séquence animée.

L'idée maîtresse est de jouer entre les composantes statiques et dynamiques, entre image et séquences animées et trouve un certain écho dans le besoin qu'ont éprouvé plusieurs artistes pour représenter le mouvement en peinture ou en photographie.

Concrètement, l'objectif de ce stage est de proposer des visualisations alternatives de vidéo en travaillant sur les composantes spatiale et temporelle de manière à interpréter rapidement l'action d'une scène audiovisuelle.

Nous sommes alors en droit de nous questionner sur la manière d'arriver à décrire l'action d'une séquence vidéo en un « coup d'œil ». Comment profiter du travail scientifique et artistique autour de la notion de mouvement ? Dans quelles mesures de telles images sont- elles pertinentes pour l'analyse rapide d'une scène ? Comment orienter par la suite la recherche de ce thème de visualisation ?

Image 1 : Etienne-Jules Marey, décomposition d'un saut périlleux…

I. INA : Institut National de l'Audiovisuel

A. Présentation

1. Historique

L'Institut National de l'Audiovisuel, créé par la réforme de l'audiovisuel menée en 1974, est né le 6 janvier 1975. Par la loi du 7 août 1974, le législateur confirme la place centrale de l'audiovisuel dans la vie sociale, culturelle et économique de la France. L'**ORTF** s'éclate en six sociétés autonomes qui prennent en charge la production (**SFP**), la programmation (**TF1**, **Antenne 2**, **France 3**, **Radio France**) et la diffusion (**TDF**).

La septième, l'**INA**, accueille des fonctions transversales de l'ORTF, et notamment le service de la recherche de l'ORTF, créé et dirigé par **Pierre Schaeffer** depuis 1959. Ce pionnier de la radiotélévision française, inventeur de la musique concrète, est le père spirituel de l'Ina.

- **Dates clés :**

1974 : *loi du 7 août* relative à la liberté de communication. Réforme de l'audiovisuel. Création de 7 sociétés de service public issues de l'**ORTF**.
1975 : naissance de l'Ina le 6 janvier
1981 : 1er salon **Imagina** (rendez-vous européen des professionnels de l'audiovisuel numérique), co-organisé avec le Festival de Télévision de Monte-Carlo
1982 : *loi du 29 juillet*, le système de dévolution de propriété des archives entre les chaînes publiques et l'Ina est institué.
1986 : *loi du 30 septembre,* modification du système de dévolution de propriété des archives qui place l'Ina dans un système concurrentiel. Création d'Ina Entreprise, filiale commerciale de l'Ina.
1987 : privatisation de **TF1** et création de **La Cinq**, **M6**, **La Sept**, sans obligation légale vis-à-vis de l'Ina.
1992 : *loi du 20 juin,* l'Ina a pour mission le dépôt légal des programmes radiodiffusés et télédiffusés.
1995 : 1er janvier création de l'Inathèque de France qui gère le dépôt légal des programmes des chaînes de **Radio France** et des diffuseurs nationaux hertziens de télévision.
1998 : l'Inathèque ouvre son centre de consultation à la Bibliothèque nationale de France.
2000 : l'Ina signe avec l'Etat le premier Contrat d'objectifs et de moyens de l'audiovisuel public.
2001 : le Festival de Télévision de Monte-Carlo devient le seul détenteur des droits sur la manifestation Imagina.

2. Statut

L'INA est un **E**tablissement **P**ublic à caractère **I**ndustriel et **C**ommercial (**EPIC**). L'Etat fixe le cadre général, législatif, réglementaire et financier dans lequel l'Ina assure ses missions. Il dispose du pouvoir de contrôle sur ses activités : présence de représentants de l'Etat et de parlementaires au Conseil d'Administration, questions parlementaires, enquêtes et rapports.

3. Missions

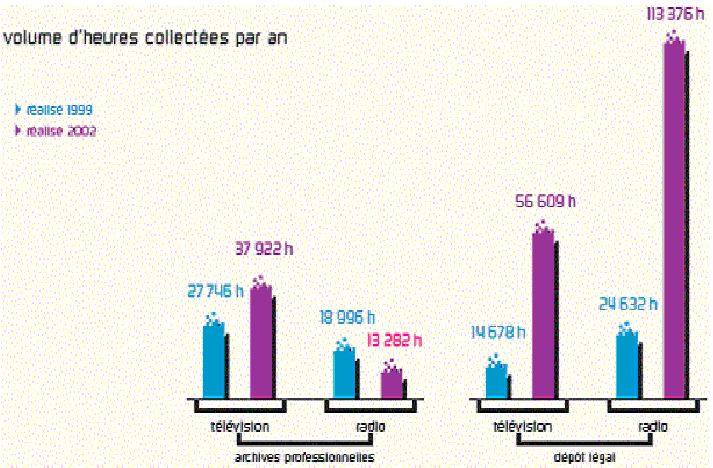
L'Ina a pour objectif de devenir « le premier opérateur européen spécialisé dans le patrimoine audiovisuel dans le nouvel environnement numérique ». La loi n° 2000-719 du 1er août 2000, modifiant la loi n° 86-1067 du 30 septe mbre 1986 relative à la liberté de la communication, précise dans son article 10, les missions de l'Ina :

- **La conservation du patrimoine audiovisuel national**
  - Assurer la collecte des programmes audiovisuels,
  - Préserver et restaurer les fonds,
  - Offrir des services documentaires renouvelés et efficaces,
  - Renforcer l'accessibilité aux images et aux sons dans l'environnement Internet.

- **L'exploitation et la mise à disposition de ce patrimoine**
  - Développer l'exploitation commerciale des fonds,
  - Valoriser les archives à des fins scientifiques, éducatives et culturelles.

- **L'accompagnement des évolutions du secteur audiovisuel à travers ses activités de recherche, de production et de formation**
  - Renforcer la convergence des activités de recherche et expérimentation vers la mission patrimoniale,
  - Accroître le caractère innovant de la production de création et de recherche,
  - Orienter la formation professionnelle vers les technologies numériques.

4. Quelques chiffres



1ère banque mondiale d'archives numérisées,

1 100 000 d'heures d'archives professionnelles fournies par les chaînes publiques de télévision et de radio depuis 1945,

930 000 d'heures de dépôt légal depuis 1995 fournies par l'ensemble des chaîne de télévision, et par 12 chaînes du câble depuis 2002.

Figure 1 : volume horaire numérisé en 1999 et 2002

5. Organigramme

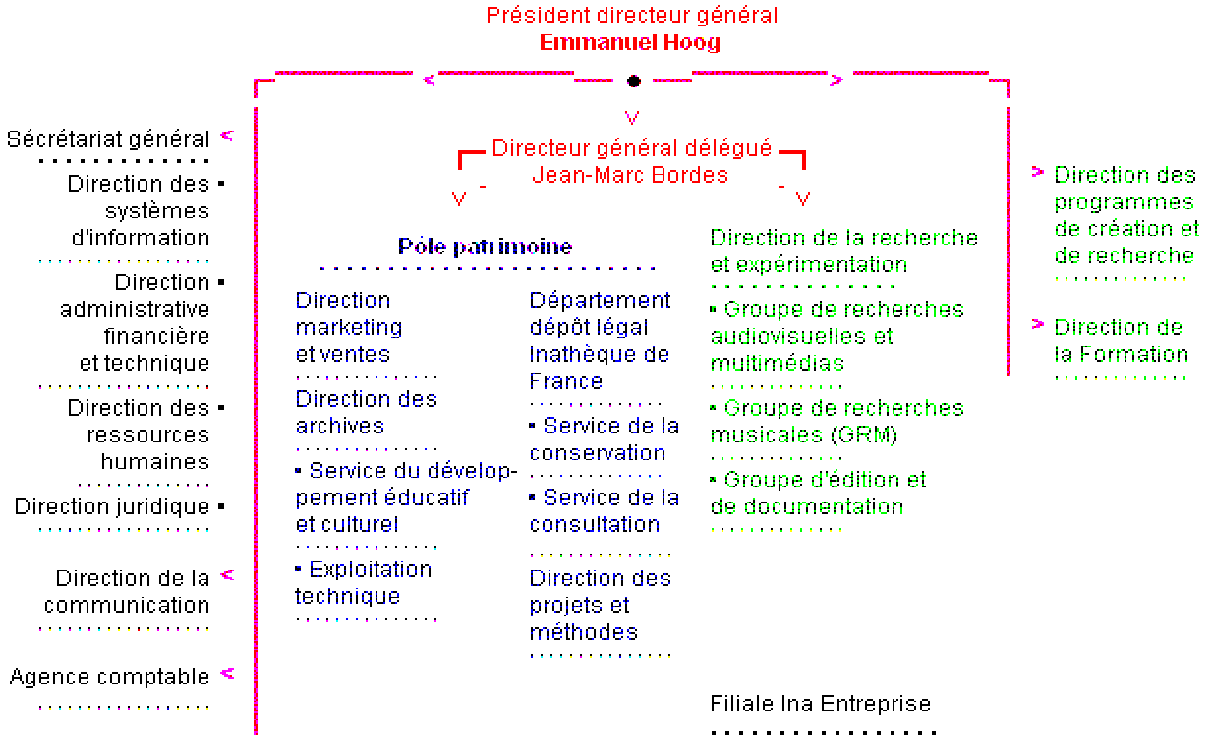


Figure 2 : organigramme de l'INA

B. Département Recherche et Expérimentation

Hérité de l'initiative de Pierre Schaeffer, l'activité Recherche de l'INA se divise aujourd'hui en 2 centres :

- le **GRM** : Groupe de Recherche Musicale installé dans les locaux de la maison Radio France, lieu de rencontre de création, de recherche et de conservation dans les domaines du son et des musiques électroacoustiques,
- le **DRE** : Département Recherche et Expérimentation à Bry-sur-Marne se subdivisant lui-même en plusieurs équipes travaillant sur des thèmes complémentaires relatifs à l'Audiovisuel.

1. Thèmes de recherche (non liés au stage)

- **Description des Contenus Audiovisuels :** DCA a pour objectif de mener des recherches et des expérimentations de nouveaux moyens d'exploitation des contenus audiovisuels en s'appuyant sur la manipulation de leur description.

- **Etude socio-économique des médias :** d'un point de vue technique, il s'agit de définir des méthodologies d'évaluation des programmes en fonction de leurs implications économique, culturelle, etc.

- **Interprétation sémiotique de l'audiovisuel :** son objectif est d'étudier les formes visuelles et sonores en tant que dimension spécifique du langage audiovisuel.

2. Thèmes de recherche concernés par le stage

a) Traitements Techniques de l'Audiovisuel (TTA)

TTA a pour vocation de fédérer les initiatives de recherche liées au traitement des images animées (support vidéo) et des sons.

Les thèmes étudiés sont principalement :
- la **sauvegarde** des documents analogiques audiovisuels : vidéo, audio, film,
- la **restauration** numérique de programmes de télévision (programmes d'archives vidéo et films),
- le suivi et la **protection** des documents audiovisuels grâce aux techniques de signatures, et de marquage invisible des images ("watermark").

b) Visualisation Interaction Expérimentation (VIE)

Il s'agit de mobiliser des techniques (mêlant infographie et 3D), pour la conception d'interfaces pour les bases de données audiovisuelles. Ces techniques doivent permettre de maîtriser la profusion et la complexité des informations audiovisuelles et de leur "métadonnées" (documentation). Ces outils doivent également se prolonger en instruments de réécriture et de réédition afin de permettre à un utilisateur de s'approprier les contenus et de les intégrer dans ses propres chaînes de valeur ajoutée.

- **Visualisation** :

Recherche théorique et développement de points clés innovants :
- pour la représentation et l'exploration de bases de documents indexés,
- et pour la représentation et l'exploration d'un document audiovisuel.

- **Interaction** :

Recherche théorique et développement d'outils pour permettre à l'utilisateur de naviguer dans les systèmes de visualisation, de rechercher une information, d'analyser un ensemble d'informations, d'assimiler progressivement l'organisation et le contenu d'un grand ensemble de données.

- **Expérimentation** :

Des collaborations avec des experts documentaires de l'Ina, ainsi qu'avec le LIRMM, permettent d'expérimenter et d'évaluer ces interfaces auprès d'un public de professionnels et de non spécialistes.

C'est en collaboration de ces deux dernières équipes que s'est déroulé ce stage faisant appel aux spécialités de traitement d'images et de visualisation.

II. Projet VISA

A. Présentation

Ce stage constitue la brique de départ d'un projet plus vaste qui n'en est qu'à ses premiers balbutiements.

Il s'agit de savoir, dans un premier temps, dans quelle mesure il est possible de récupérer la technologie de signature d'image développée par l'équipe TTA pour créer des objets audiovisuels particuliers.

Image 2 : David Hockney, portrait de sa grand-mère

Le projet s'inscrit dans la problématique de la visualisation de gros volume vidéo. Un utilisateur recherchant des données audiovisuelles peut se retrouver dans deux cas de figure :

- soit il a une idée précise du document désiré (titre, année, auteur, …),
- soit il « butine », il ne connaît pas l'existence du document, mais recherche des documents abordant un thème (images de guerres, du sport,...), tout en restant sur d'autres médias transversaux.

Dans les deux cas, il faut vérifier que le document sélectionné est le bon ou qu'il correspond au thème. Il est donc indispensable de visionner un extrait vidéo.

La démarche classique consiste à avoir un lecteur en ligne avec une barre de lecture permettant de naviguer dans la vidéo rapidement. Mais le risque d'une telle démarche est d'épuiser et d'énerver le consultant devant un travail aussi fastidieux.

L'objectif du projet est de proposer un outil innovant de visualisation de document audiovisuel qui tente de raccourcir les temps de consultations. Le but est alors d'arriver à une « compression » temporelle et spatiale de l'image par :

- la mise à plat des images successives,
- l'ajout d'informations dans les images exprimant le temps, le mouvement et la mise en scène.

L'idée maîtresse est donc de passer de données dynamiques vers des données statiques enrichies.

- **Exemple**

Le cas d'école est la mise à plat d'un travelling. Le résultat désiré est pouvoir coller les différentes « frames » en faisant ainsi apparaître une seule grande image.

- **Dualité d'idées …**

En simplifiant la problématique, nous pouvons opposer ces deux idées :

- le processus mis en œuvre réduira une séquence vidéo, soit n images consécutives, à une seule image et le gain de temps est flagrant,
- mais une vidéo occupe une surface limitée dans une interface graphique, alors qu'un travelling optique mis à plat peu prendre rapidement des proportions énormes.

Après la mise en œuvre d'un tel outil, il est donc indispensable d'effectuer une étude qualitative et de discuter de l'impact de ce type d'images pour établir les fondements du projet plus vaste de l'équipe VIE qui prolongera ce stage.

- **intérêt pour les deux équipes**

A plus long terme chaque équipe devrait pouvoir profiter de cette réalisation :

- TTA : l'appariement de points d'intérêt et la technique pour recaler les images doit permettre de repérer les différents déplacements dans une scène pour calculer des signatures « mouvement » complétant et affinant ainsi le système de «monitoring»,

- VIE : réaliser une interface innovante de visualisation de vidéos mises à plat temporellement et spatialement, et poursuivre des recherches plus approfondies sur ce thème.

Une première étape de ce stage consiste à établir un état de l'art afin de prendre connaissance des travaux d'autres équipes dans le monde, de pouvoir profiter de théories, d'algorithmes et, d'idées, publiés par la communauté scientifique.

Cette démarche permettra de mettre en œuvre plus rapidement un premier prototype. Mais avant tout, il est nécessaire d'assimiler la technique de signature développée par TTA afin de repérer quels articles seront pertinents dans notre cas pour aboutir à un mini-éditeur.

B. Projet « monitoring »

1. Problématique

L'objet de ce projet de recherche répond à un besoin de contrôle et de surveillance des droits de l'image. En effet, malgré une législation assez stricte sur les droits de l'audiovisuel, il arrive régulièrement que des images soient exploitées sans autorisation. Le système « monitoring » propose une alternative au «watermarking» dans laquelle aucune information n'est ajoutée au média.

2. Principes

La technologie de signature d'images développée à l'INA [JFB 03] tente de réduire au maximum les données relatives à une séquence vidéo pour en faciliter le stockage et la comparaison. Le but est de reconnaître dans un temps compatible avec le temps réel des séquences d'images au sein d'un très grand ensemble de vidéos (au moins 100 000 h). Le principe consiste à détecter des points d'intérêt et de les associer à des données, des « descripteurs », constituant alors une « carte d'identité » de l'image

- **Détecter des « coins » :**

Un « coin » est un point d'intérêt de l'image, c'est-à-dire un pixel autour duquel la luminance varie fortement en intensité. Un coin implique donc un taux d'information important. L'ensemble de ces coins donne plusieurs positions caractéristiques de l'image et constitue alors une base pour calculer des descripteurs.
De plus, pour que la technologie soit performante, il est impératif quelle soit **robuste**, c'est-à-dire quelle soit capable de repérer les points d'intérêt malgré une transformation de l'image comme un changement d'intensité, une rotation, un changement d'échelle, ou une translation.

- **Calculer des descripteurs :**

Chaque point d'intérêt est associé à un descripteur qui n'est ni plus ni moins qu'un vecteur, et l'ensemble est stocké dans une base de données consultable lors d'une recherche de documents audiovisuels.
Il ne s'agit donc pas de « watermarking» : la technologie n'ajoute pas d'informations dans l'image. La vidéo est indexée, soit associée à un ensemble de signatures pour quelques images déterminées.

- **Organiser et comparer les signatures :**

Ces signatures sont organisées afin de permettre une consultation rapide de la base de données. Le choix des outils mathématiques et les méthodes de recherche de signatures sont primordiaux pour assurer :

- la robustesse au changement de prise de vue,
- la rapidité de la requête,
- l'efficacité du système.

- **Remarques :**

Un prototype est en cours de validation signe 300 heures par jour et permet (sur un PC standard cadencé à 3GHz avec 2Go de RAM) de :

- calculer à 12 fois le temps réel les signatures,
- rechercher une séquence en 2 fois le temps réel dans 300 000 heures.

3. Déroulement

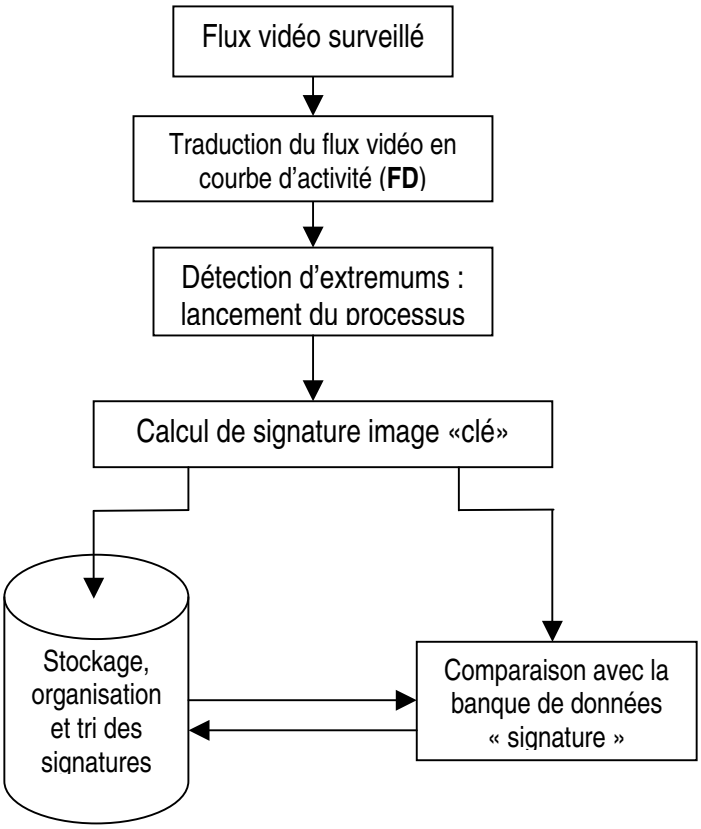


Figure 3 : les étapes du système de monitoring

a) Traduction du flux vidéo en courbe d'activité

Il est inutile d'analyser chacune des images de la vidéo : les données deviendraient très vite faramineuses et redondantes. Il faut alors analyser uniquement des images « clés », sous-entendant une variation importante de luminance (migration ample de pixels, changement d'éclairage, «cut»). Le système traduit le flux en courbe d'activité déduite par une différence en luminance d'image à image ou «FD » (Frame Difference).

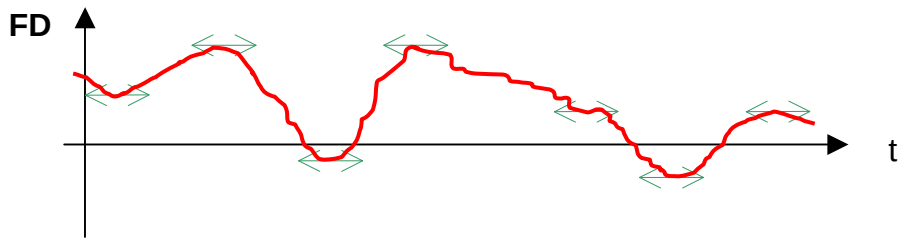


Figure 4 : courbe d'activité en luminance d'une séquence vidéo

Le calcul de signature est lancé à chaque extremum, ou toutes les 3 secondes si aucun pic n'a été détecté.

b) Sélection de points d'intérêt par le détecteur Harris

Un point d'intérêt concentre beaucoup d'informations sur son voisinage. Ce « coin » maximise donc les valeurs de dérivées autour de lui. En effet, un pixel d'une zone totalement uniforme, ne contenant aucun changement de luminance, voit ses dérivées tendrent vers zéro.

Plusieurs méthodes de détection de points d'intérêt ont été étudiées par la communauté scientifique comme le carré du gradient, l'utilisation d'un Laplacien ou la fonction de Lowe [MC 01]. Le filtre de Harris, basé sur le gradient, s'est imposé comme une des méthodes les plus efficaces ; il est notamment robuste aux rotations [HAR 84]. Ce filtre passe sur toute l'image et calcule pour chaque point un vecteur contenant le résultat de calculs de dérivées à son voisinage. Un seuillage permet de ne retenir que les points les plus porteurs d'information, qui maximisent les valeurs des dérivées.

c) Calcul de descripteurs

Un descripteur associé à un coin est, dans ce système, un vecteur à 9 coefficients :

- les dérivés en x, y et xy du 1er ordre,
- les dérivés en x et en y du 2nd ordre,
- 4 positions voisines.

Ces positions interviennent pour une opération de recentrage ou pour palier au problème de changement d'échelle. En effet, lorsque que le système se concentre sur une petite fenêtre autour du point d'intérêt, il peut faire des confusions avec le bruit présent dans l'image. La solution retenue est d'effectuer d'abord Harris sur une grande fenêtre autour du point d'intérêt, de retenir 4 autres points voisins, puis d'effectuer à nouveau Harris dans une fenêtre plus petite en se recentrant sur le coin grâce à ces points environnants.

A la fin des calculs, les descripteurs sont triés selon leur taux en informations et rangés par ordre décroissant dans une liste. Au final, toutes les informations de signature sont stockées dans une base de données : à chaque image clés de la séquence vidéo correspondent en 2 tableaux : le premier contenant les coordonnées des coins et le deuxième les vecteurs descripteurs associés.

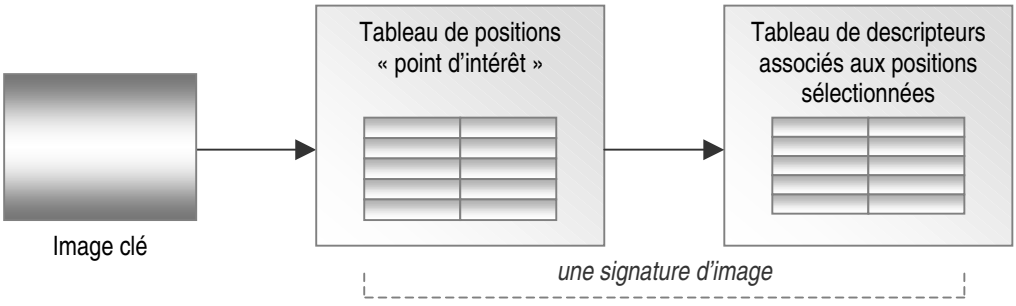


Figure 5 : distinction des données en jeu durant le "monitoring"

Remarque : le volume de données créées est nettement inférieur à celle de la vidéo.

d) Stockage et organisation des signatures

L'ensemble des signatures est organisé selon un modèle fractal établit grâce à une courbe de Hilbert, afin d'optimiser les temps d'accès et de comparaison entre descripteurs. En effet, pendant la recherche de signatures les têtes de lecture des disques durs parcourent des distances importantes. Le modèle fractal trie et rapproche les descripteurs « voisins » issus de différentes vidéos. Ainsi, lors d'une requête, le parcours effectuer par la tête de lecture est nettement raccourci ce qui diminue et améliore considérablement les temps de recherche de signature [JFB 03].

e) Comparaison des signatures avec la banque des images propriétaires

Pour rechercher une séquence vidéo à partir d'une image, il suffit de comparer les signatures à celles stockées dans la base de données. La comparaison ne se fait pas d'une image à une autre mais sur un ensemble d'images clés successives englobées dans une durée d'environ 3 secondes. Une fenêtre glissante large de cette durée passe sur les signatures associées et effectue un calcul de distance entre les descripteurs « candidats » et les descripteurs « requêtes ».

Le critère d'évaluation est une simple distance entre les vecteurs signatures. La distance euclidienne dans un espace à 9 dimensions constitue un bon compromis entre temps de calcul et en efficacité.

$$L_2 = \sqrt{\sum_{i=0}^{n} (yi - xi)^2}$$

yi : élément de la signature candidate
xi : élément de la signature recherchée
ici avec n = 9, la taille des descripteurs

Deux images « proches » minimiseront alors la somme des distances entre leurs descripteurs. La décision de retenir l'image clé correspondante à l 'image « requête » se fait à posteriori sur un ensemble d'images clés pendant 3 secondes, ce qui contribue à une certaine fiabilité puisque l'image est considérée dans son contexte.

C. Prospection : accroche des objets d'une scène

Dans ce sujet, nous nous basons sur trois hypothèses sur un couple d'images successives d'un même plan :

- elles sont fortement corrélées,
- les variations de luminances sont faibles,
- les points d'intérêt peuvent à priori être suivis.

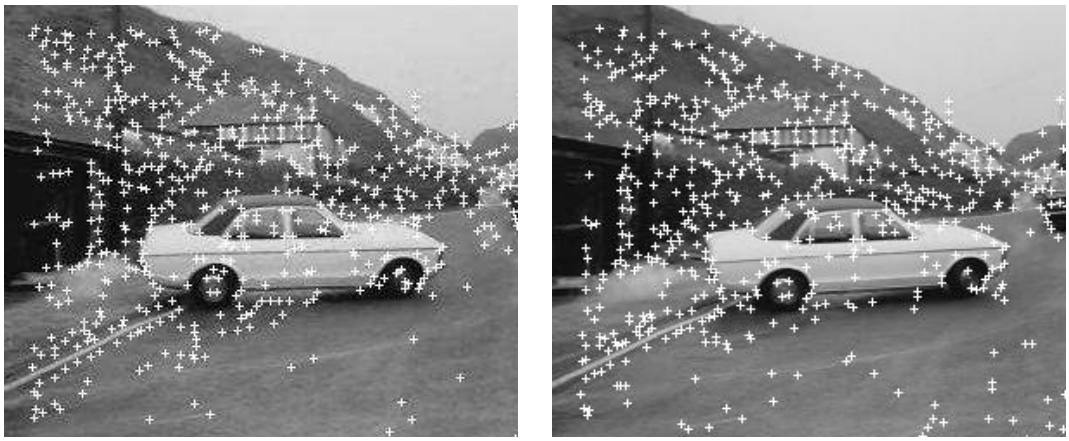


Image 3 : détection de points d'intérêt avec le filtre de Harris sur 2 images consécutives

Nous pouvons observer sur cet exemple l'accrochage de certains points notamment dans le fond, sur la colline, mais aussi sur l'objet en mouvement par rapport au cadre : sur les roues et les bords des carrosseries de la voiture. Il est donc légitime de penser qu'il est possible de matcher les coins et de déterminer plusieurs vecteurs mouvements composant l'image :

- ceux liés au mouvement global, celui du cadre,
- ceux liés aux mouvements locaux des éléments de la scène.

III. Etat de l'art et discussion

Plusieurs travaux sur le résumé vidéo existent et s'appuient sur plusieurs thèmes :

- l'appareillage d'images,
- la théorie des estimateurs et des votes appliquée en vision,
- la segmentation d'objets audiovisuels,
- les transformations géométriques.

Dans une certaine mesure, nous pourrions aborder d'autres thèmes liés à la culture cinématographique et à l'interprétation de l'Image comme :

- la sémiologie graphique (ajout d'informations dans les images, significations implicite et explicite),
- la grammaire filmique et l'écriture de « storyboard ».

Nous nous limiterons ici aux aspects techniques et théoriques.

A. Etat de l'art

1. Estimation robuste de paramètres en vision par ordinateur

Les applications en vision par ordinateur font régulièrement appel à des procédés d'estimation de paramètres. Charles V.Stewart ([Ste 99]) propose un inventaire de différentes techniques d'estimation robuste de paramètres afin de distinguer les « bonnes » données («inliers») des erreurs («outliers»).

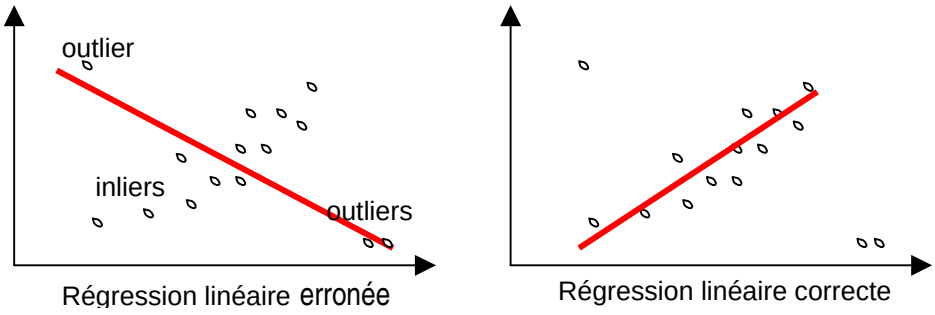


Figure 6 : cas d'une régression linéaire

Le principe est de faire émerger d'un ensemble de données un caractère modélisé par une fonction. En effet, toute expérimentation contient des résultats erronés (imprécision, aberration,…), et une modélisation robuste doit pouvoir faire le tri entre les données correctes («**inliers**»), et les données incohérentes («**outliers**»). La Figure 6 illustre le cas d'une régression linéaire : la fonction de gauche n'est pas suffisamment robuste aux «outliers» et modélise cet ensemble de points par une droite non significative.
Nous pouvons distinguer deux principales familles d'estimateurs :

- les « M-estimateurs »,
- les techniques des moindres carrés (LMS: Least Median of Squares), qui se déclinent en diverses méthodes utilisés en vision par ordinateur : RANSAC, MSAC, MUSE, ALKS, MINPRAN…

Les M-estimateurs utilisent un procédé itératif de pondération (IRLS: iteratively reweighted least squares »). La modélisation optimale est approchée en plusieurs étapes d'estimation. A chaque itération, des poids sont réévalués pour faire converger le modèle. La pondération des données est effectuée par des fonctions de coût, et dont 3 sont des références dans le domaine de la vision :

- Beaton and Tukey considérée comme provocant la réjection d' «outliers» la plus agressive,
- Cauchy,
- Huber.

La deuxième famille d'estimateur est plus directe : le procédé implique de multiples essais avant d'atteindre un solution optimale. Stewart démontre l'efficacité d'un IRLS combiné avec Beaton and Tukey sur un LMS lors d'une régression linéaire sur un jeu de données pseudo-aléatoires : la première converge moins rapidement mais est plus précise qu'un LMS.

- **Applications :**

Comme dans tout problème, une technique est plus adaptée qu'une autre selon le contexte et l'application particulière. Par exemple, Smolic, Sikora et Ohm ([SSO]), dans leur algorithme d'appariement de zones d'intérêt entre images, préfèrent utiliser un M-estimateur plus sensible au match «outliers» qu'un LMS.

2. Estimation de mouvements

a) Modèles potentiels

Stiller et Konrad dans [SK99] recensent les différents modèles pouvant être appliqués en vision (Tableau 1).
Seul le cas de la projection linéaire satisfait le modèle de caméra perspectif donnant une impression de profondeur de la scène. En effet, le modèle orthographique ne possède pas de ligne de fuite et entraînera une déformation de la scène reconstruite.

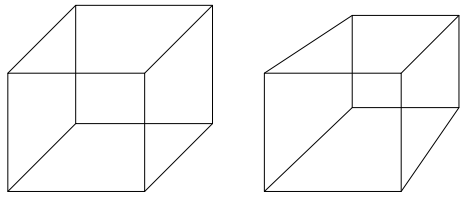


Figure 7 : Modèle de caméra

Remarque : la cas polynomial est caractérisé par un modèle de caméra arbitraire et offre (peut être ?) la possibilité de se placer dans un repère perspectif (ce point est discuté dans la partie expérimentation, au paragraphe V.B.3.a).

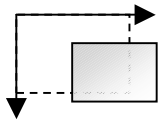
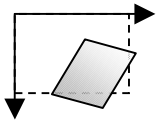
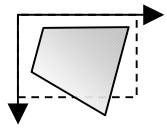
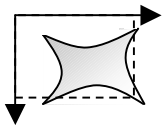
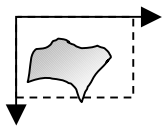
Modèle 2D				Modèle caméra
-	Coefficients	Equations	Illustration	
Translation	2x1	$x = x' + t_x$ $y = y' + t_y$		Orthographique
Affine	2x3	$x = \frac{\cos\theta}{Sx} x' + \frac{\sin\theta}{Sx} y' + t_x$ $y = -\frac{\sin\theta}{Sy} x' + \frac{\cos\theta}{Sy} y' + t_y$		Orthographique
Projection linéaire	2x4	$x = \frac{ax' + by' + c}{mx' + ny' + p}$ $y = \frac{dx' + ey' + f}{mx' + ny' + p}$		Perspectif
Quadratique	2x6	$x = a_0 + a_1 x' + a_2 y' + a_3 x' y' + a_4 x'^2 + a_5 y'^2$ $y = b_0 + b_1 x' + b_2 y' + b_3 x' y' + b_4 x'^2 + b_5 y'^2$		Orthographique
Polynomial	2xn	$x = \sum_{i=0}^{n} c_n x'^n y'^n$ $y = \sum_{i=0}^{n} d_n x'^n y'^n$		Arbitraire

Tableau 1 : les transformées utilisée pour une estimation de mouvement

3. Images mosaïques

a) Approche imagerie - robotique - vision

Il existe de nombreux travaux sur l'appariement et la fusion d'images dans les domaines de l'imagerie médicale, la géodésie, la reconstruction de scène 3D, et autres applications scientifiques.

Mais ce type de travaux est plus de l'ordre de l'instrumentation : la plupart des articles exploitent un matériel d'acquisition calibré et spécifique pour une situation particulière de mouvement de caméra.

Certes, les résultats proposés sont souvent excellents, car les systèmes sont optimisés pour un cas précis, mais, idéalement, nous devons traiter toutes les situations :

- de mouvement de caméra (travelling, rotation, travelling optique…),
- de prise de vue (tout objectif, support argentique, vidéo, …),
- tous les types d'images (animation, 3D, images naturelles, etc).

Néanmoins, la rigueur scientifique de ces documents illustre l'efficacité de certains procédés de reconstruction d'images. Ainsi, dans le cas particulier de la reconstruction de réseau rétinien en imagerie médicale, selon [Lal 01], le modèle polynomial n'apporte point de qualité en plus par rapport au modèle affine (estimée à 65% contre 66%).
Une bonne illustration de ce type de modèles concerne le panorama cylindrique et sphérique.

(1) Panoramas cylindriques et sphériques

Le panorama cylindrique crée une mosaïque selon un modèle de caméra perspectif. Les panoramiques sont souvent réalisés avec des objectifs grand angle, ce qui induit une grande profondeur de champ. Cet effet est encore accentué si l'objet focalisé est éloigné de la caméra. Les objets au centre donnent l'impression d'être plus petits (voir V.A.2, « Rappel sur les notions d'angle et de profondeur de champs » p.46). En conséquence, un modèle de transformation « projection linéaire » réintroduit toute cette perspective dans l'image mosaïque. Ainsi, comme l'illustre David Capel et Andrew Zisserman [CZ], un panoramique verra son centre fortement aminci, alors que la projection sur un cylindre formera une image rectangulaire et incurvant les lignes droites.

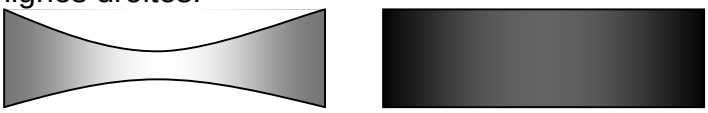


Figure 8 : modèle projectif de la mosaïque reconstruite d'un panoramique, et sa projection cylindrique

J'effectue une courte analyse sur ce type d'effet dans la partie V.B.3.a).
Le problème de cette technique est qu'elle nécessite une connaissance préalable des réglages de la caméra comme le rappel [CZ], ce qui nous ramène à la même problématique que le paragraphe précédent.
Par extension, le panorama sphérique induit les mêmes considérations techniques de prises de vue.

Remarque : ce point nous éclairci mieux sur la rapidité des logiciels de « Stiching » fournis avec la plupart des appareils photo-numériques. Tous les paramètres de calibrage sont connus et peuvent utiliser ce type de modèle.

Richard Szeliski et Heung-Yeung Shum [SHZ] proposent une solution pour contrer ces insuffisances. Il suffit de poser la caméra sur un tripode et de connaître la longueur de la focale. Le tripode consiste à compenser les problèmes de parallaxes

lors d'un mouvement de rotation de la caméra. Le capteur CCD coïncide alors avec le centre de rotation de la caméra.
Ce procédé ne peut être appliqué qu'à un seul type de plans. Nous ne pouvons demander à tous les techniciens de l'audiovisuel d'ajouter un tripode sur chaque pied de caméra ! En revanche, cette méthode est tout à fait envisageable sur commande pour le domaine du multimédia.

(2) Panoramas en modèle rotation

C'est la méthode explorée par Szeliski et Shum [SHZ] pour réaliser des panoramiques « pleins ». Le principe consiste à éliminer 5 paramètres d'un modèle homographique pour se retrouver avec une matrice de transformation à 3 paramètres. Il est intéressant de voir que cette technique s'inspire de modélisation 3D, et que la focale est prise en compte sous forme matricielle :

X~TVRp avec p le point de l'espace, x son image, T la matrice translation, V la matrice exprimant un facteur d'échelle de la longueur de focale et R la matrice de rotation 3D.

$$T = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \quad V = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ le but étant d'optimiser R

(3) Panoramas de routes

Nous pouvons encore citer le travail de Jiang Yu Zheng [Zhe03] qui reconstitue des panoramas à partir de prise de vue d'une caméra embarquée sur un véhicule. Il propose des méthodes de compensation des problèmes liés aux virages, aux effets d'occlusions sur des éléments minces comme les troncs d'arbres et les poteaux, et au problème de trajectoire en zigzag d'une voiture. Il emploi un modèle de projection orthogonal et connaît préalablement la longueur de focale.

b) Modélisation sans contraintes de calibrage

(1) Modèle polynomial et estimation hiérarchique

Stewart dans son article sur les estimateurs [Ste 99] propose une approche hiérarchique de l'estimation de transformées d'images afin de reconstruire des images mosaïques par un modèle polynomial (réduit en réalité à un modèle quadratique). Il affirme que déterminer directement ces 12 coefficients à partir d'une vaste disponibilité de points « matchée » est trop approximatif.

Algorithme :

1.estimer un modèle translation selon un critère de corrélation d'images et retenir toutes les correspondances valides pour cette transformation,
2. estimer à partir de ces correspondances un modèle affine par un estimateur LMS,
3.estimer le modèle quadratique en implémentant de manière itérative (IRLS) un M-estimateur : la première itération part du modèle affine précédant.

L’estimation des transformées se fait à partir des «features», ou caractéristiques de l’image. Ce sont par exemple les contours, les points d’intérêt, et leur choix est souvent intimement lié à l’application de la reconstruction de mosaïques.

Stewart illustre sa méthode par la reconstruction d’un fond de rétine humaine (imagerie médicale) et base son estimation de transformées sur l’extraction des caractéristiques des réseaux sanguins de l’œil : les vaisseaux sanguins sont détectés et symbolisés par des lignes blanches.

Dans [SSO 99], les auteurs combinent un «feature matching» avec des techniques de flux optique, en s’inspirant des filtres de Kalman, pour établir des modèles polynomiaux. Cet article nous averti sur le manque de souplesse de la technique : ces équations paraboliques ne peuvent être inversées. En conséquence, à chaque étape d’appareillage, les paramètres de mouvements sont réactualisés petit à petit, en dynamique. Autrement dit, après avoir analysé toute la séquence d’image et trouvé toutes les transformations quadratiques, il est impossible d’exprimer une transformation dans un autre repère.

(2) Algorithme RANSAC

L’algorithme RANSAC (Random Sample Consensus) revient dans plusieurs articles, et semble être efficace pour estimer la transformation entre deux images proches. RANSAC est une méthode d’ajustement robuste de modèle malgré une grande présence d’»outliers» [FB 81]. Plusieurs auteurs le citent pour créer des images mosaïques.

Philipe Hans Torr et Andrew Zisserman [TZ 99] proposent une application de RANSAC pour réaliser des relations d'appariement d'images sans passer par l’analyse de la structure de la scène, seulement avec le mouvement de la caméra. L'homographie à déterminer est représentée par une matrice 3x3 transformant les coordonnées homogènes de l'image : x' = Hx.

Algorithme :

1. détection de points d'intérêt
2. calcul d’un jeu de correspondances de points d'intérêt, basé sur la proximité et la similarité de l'intensité entre pixels voisins
3. estimation robuste RANSAC:
    a. sélection aléatoire de 4 correspondances et calcul d’une homographie H
    b. calcul la distance erreur géométrique de l'image pour chaque jeu de correspondances
    c. calcul le nombre d' «inliers» non contradictoires avec H par correspondance où la distance d'erreur est inférieure à un seuil. Retient alors H qui possède le plus grand nombre d' «inliers».
4. re-estimation de $H_{optimal}$ avec toutes les correspondances recensées comme «inliers» en minimisant le coût maximum d’une fonction de ressemblance. Utilisation d’une suite numérique minimisée (algorithme Levenberg-Marquardt).
5. appariement guidé : d'autres correspondances de points d'intérêt peuvent être maintenant déterminées selon $H_{optimal}$ pour définir des régions de recherche.

La méthode d'estimation robuste est essentielle dans l'algorithme car 40% des correspondances sont incorrectes selon leurs expérimentations. Nous pouvons étendre ce procédé liant deux images pour la construction de mosaïques :

- calcul des points d'intérêt,
- calcul des homographies,
- estimation de l'homographie optimum,
- assemblage des images en une.

L'homographie optimale est paramétrée pour fonctionner sur l'ensemble des images. Par exemple, l'homographie entre les images 1 et 3 se compose de celles entre la 1 et 2, puis entre la 2 et la 3 : $H_{13} = H_{23} * H_{12}$.

Applications : résumé vidéo, **super résolution** (à partir de plusieurs clichés contenant une zone commune, il est possible d'augmenter le détail sur cette région), et éventuellement pour la restauration.

(3) Motion panoramas

Les recherches les plus clairement orientées « audiovisuelles » sont proposées par Bartoli, Dahal et Horaud [BDH 03]. Leur projet est de créer des panoramas de mouvements à partir de caméra non-calibrée effectuant un mouvement de **rotation pure**. L'idée est de reconstruire une image mosaïque en alignant avec précision les images successives. Cette précision permet alors la reconstruction du fond statique et une segmentation des objets dynamiques. Ils illustrent leurs exemples sur des séquences d'athlétisme où est visible la décomposition des mouvements.
Leur **algorithme** se divise en 4 étapes :

1. appareillage de points d'intérêt,
2. estimation grossière du mouvement avec l'algorithme robuste MSAC (hérité de RANSAC),
3. raffinement de l'homographie par l'algorithme Levenberg-marquardt,
4. alignement des images par une méthode directe basée sur les intensités.

Cet algorithme se distingue de [CZ] par la limitation du modèle à une rotation pure. Le principe est d'analyser l'ensemble des images appareillées pour faire une correspondance globale sur tous les points de même rayon dans l'espace 3D.
Un point essentiel de leur technique est que le cas de rotation pur permet de simplifier le modèle homographique à un modèle à 4 paramètres.
Ainsi,

$$x = \frac{ax'+by'+c}{mx'+ny'+p} \qquad y = \frac{dx'+ey'+f}{mx'+ny'+p} \quad \text{devient} \quad x = \frac{t_x}{mx'+ny'+1} \qquad y = \frac{t_y}{mx'+ny'+1}$$

en excluant les zooms et la rotation par rapport au centre de l'image.

B. Discussions

1. Corrélation avec les techniques de réalisation

En terme de techniques de captation vidéo, il semble évident que le travelling peut être modélisé par une translation et qu'un mouvement de rotation peut l'être par une transformée affine. Mais d'autres procédés de réalisation audiovisuelle sont à éclaircir.

a) Confusion entre zoom et « scale »

Premièrement le travelling optique ou zoom est modélisé de manière imprécise par une transformation affine, malgré une composante liée à l'échelle. En effet, cette transformée ne considère que des mouvements rigides. Or, le zoom effectue une migration des pixels selon des lois quadratiques.

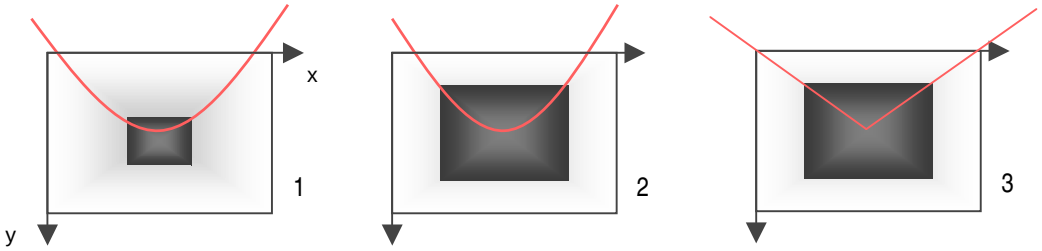


Figure 9 : illustration de la migration de pixels durant un travelling optique suivant une loi quadratique (1 et 2), contre le cas du changement d'échelle(3).

b) Confusion entre panoramique et rotation

La rotation induite par la transformée affine ne peut être assimilée à la rotation axiale de la caméra lors d'une séquence panoramique prise avec un objectif grand angle par exemple. Encore une fois le modèle affine est une transformation rigide de l'image et implique un mouvement circulaire des pixels dans l'image.

Il semble plus exact de réduire un mouvement de panoramique à un modèle de translation (nous discuterons de ce point plus loin dans le chapitre des résultats visuels).

Remarque : nous pouvons pressentir une certaine difficulté à adapter un modèle à une situation de captation lors de mouvements de caméra complexes combinant plusieurs procédés (par exemple avec une «steadycam» changeant de focale pendant la prise de vue). L'idéal serait de pouvoir s'affranchir du modèle à appliquer.

2. Choix d'une méthode de construction de mosaïques

Il est nécessaire d'employer une technique d'appariement d'image qui s'affranchit d'une analyse précise de la structure, comme les contours et la géométrie de la scène, en se concentrant sur les points d'intérêt.

Nous pouvons exclure toutes les techniques de vision robotique puisqu'elles nécessitent une connaissance préalable des paramètres de la caméra.

L'approche hiérarchique de Stewart nécessiterait d'être validée : il faudrait alors vérifier que nos «features» que sont les points d'intérêt, portent l'information adéquate pour construire ce type de modèle.

La méthode de [SSO], quant à elle combine un « feature matching » et les techniques de flux optique, ce qui rend le système plus complexe.

La méthode proposée par Philipe Hans Torr et Andrew Zisserman est retenue pour orienter la réalisation du prototype car elle est suffisamment générale pour être considérée dans plusieurs situations de cadrage. Cependant un gros travail reste à fournir pour compléter les informations partielles de la plupart des papiers ; les auteurs ne nous livrent pas tous leurs secrets.

3. Discussion sur la notion d'homographies

Le terme d' « homographie » est employé pour exprimer une transformation entre deux images, mais il reste une certaine confusion autour de sa définition. Selon François Sillon [Sil] une transformation appelée « homographie » lie deux images provenant d'une scène à partir du même point de vue (cas du panoramique). Sa formule est celle de la projection linéaire (voir Tableau 1).
Plus généralement, le terme d'homographie est assimilé aux transformations affines et projectives, et exclut donc les modèles polynomiaux.

Selon Richard Szeliski et Heug-Yeung Shum [SS 97] beaucoup d'auteurs recommandent le modèle de mouvement perspectif à 8 paramètres :

$$\begin{bmatrix} m0 & m1 & m2 \\ m3 & m4 & m5 \\ m6 & m7 & m8 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ correspondant à la transformation projective citée précédemment : $x = \frac{m0x'+m1y'+m2}{m6x'+m7y'+m8}$ $y = \frac{m3x'+m4y'+m5}{m6x'+m7y'+m8}$

En réalité nous pouvons concentrer tous les degrés de liberté de la caméra dans ce modèle :

$$\begin{bmatrix} \frac{\cos\theta}{Sx} & \frac{\sin\theta}{Sx} & tx \\ -\frac{\sin\theta}{Sy} & \frac{\cos\theta}{Sy} & ty \\ m6 & m7 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ où l'on retrouve les composantes affines

Les auteurs de [BDH 03] profitent du cas du panoramique, sans mouvement de rotation autour du point central du cadre et sans zoom (avec un mouvement affine limité à la translation donc), pour réduire le modèle homographique à 4 paramètres :

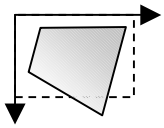
$$\begin{bmatrix} m0 & m1 & m2 \\ m3 & m4 & m5 \\ m6 & m7 & 1 \end{bmatrix}\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ devient } \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ m6 & m7 & 1 \end{bmatrix}\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

En conséquence deux couples de correspondance de points seulement suffiront pour estimer cette homographie réduite.

Torr et Zisserman [TZ 99] ne précisent pas dans leur algorithme le modèle utilisé. La matrice « homographie » peut aller de la simple translation à la projection linéaire. Leurs exemples, par contre, montrent que le modèle est projectif. Chaque équation possède 5 paramètres. Or dans l'algorithme de Torr et Zisserman, la première estimation d'homographie se fait à partir de 4 correspondances de positions.

Supposition : la première estimation d'homographie, nécessaire pour détecter les «inliers», doit être réalisée avec le modèle polynomial à 4 paramètres de la forme :

$$x = a_0 + a_1 x' + a_2 y' + a_3 x' y'$$
$$y = b_0 + b_1 x' + b_2 y' + b_3 x' y'$$



Ce modèle polynomial déforme les images en donnant un effet de perspective similaire au modèle projectif.

Remarque : il est alors légitime de penser que ce modèle polynomial pourrait convenir à la place du modèle projectif. Trois raisons s'y opposent pourtant :

- il possède moins de paramètres de contrôle (4 contre 5 pour le projectif),
- il est impossible de le représenter sous forme matricielle, et le rend incompatible avec l'algorithme de Torr et Zisserman,
- il n'y a pas de composante quadratique, ce qui rend le modèle trop réducteur dans le cas du zoom.

4. Problème d'estimation d'homographie – algorithme Levenberg-Marquardt

Plusieurs articles proposent de raffiner l'estimation d'homographie à partir de l'ensemble des «inliers» recensés avec la méthode RANSAC. Deux méthodes non-linéaires sont applicables au cas des mosaïques :

- optimisation de Newton citée dans [PM],
- algorithme de Levenberg-Marquardt ([TZ 99], [Sil]).

Dans les deux cas, l'approche est itérative : les algorithmes proposent d'ajuster des paramètres d'un modèle en minimisant une fonction de coût à chaque passage dans la boucle.

Selon [PM] la méthode de Newton généralement converge vite après seulement 2 ou 3 itérations. Levenberg-Marquart plus cité, paraît plus populaire.

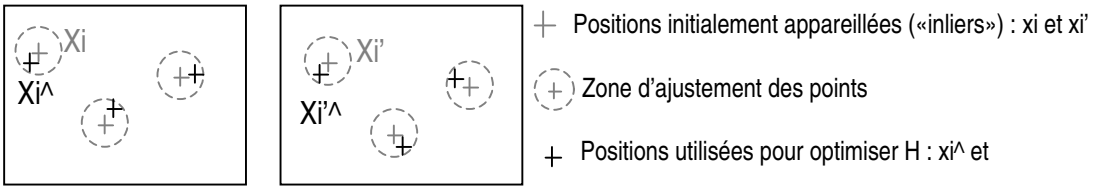


Figure 10 : optimisation par Levenberg-marquardt selon [TZ 99]

[TZ 99] proposent de minimiser le critère suivant :

$\sum_{0}^{i} d(xi, \hat{xi})^2 + d(xi', \hat{xi'})^2$, c'est-à-dire que pour chacun des «inliers» (xi et xi'), l'algorithme recherche dans leur voisinage les positions qui minimiseront cette somme. La combinaison de ces points donnera alors la transformée optimisée.

IV. Développement d'un mini éditeur

A. La démarche : une approche modulaire

1. Environnement

L'environnement de travail fourni reprend un ensemble de librairies développées en langage C par l'équipe TTA, sous Linux (distribution RedHat 9) et destinées initialement à expérimenter différents détecteurs et signatures.

Les images sont manipulées avec la librairie **Dali.** Nous travaillons sur des « ByteImage », sur une profondeur de 8 bits, en niveau de gris dans un premier temps. Les vidéos sont préalablement transformées en suite de fichiers images pgm (format brut), avec l'utilitaire **ffmpeg** par exemple.

Remarque : cette librairie sera très prochainement remplacée par **Gandalf** et permettra le traitement directement sur le flux vidéo.

La GNU Scientific Library (**GSL**) s'est greffée au cours de ce stage car elle offre des fonctions de calculs scientifiques avec des algorithmes optimisés, notamment le calcul matriciel et l'ajustement des moindres carrés,.

Cette démarche s'inscrit dans la volonté de proposer à la DRE un environnement complet de traitement vidéo, de calcul scientifique et de monitoring, avec des possibilités d'accéder à une partie de la banque audiovisuelle de l'INA.

A partir de ces librairies, le projet démarre «**from scratch**». Rien n'a été développé autour du sujet de la mise à plat temporelle de vidéo à l'INA et je dois profiter au mieux de toutes ces fonctions pour arriver à un premier prototype satisfaisant et ce malgré une mise en œuvre conséquente.

2. Orientation

Selon les conclusions de l'état de l'art, le modèle le plus complet semble être la projection linéaire. Mais étant donné que ce stage s'inscrit dans un projet plus vaste qui n'en est qu'à ces débuts, il me semble opportun de suivre une ligne directrice qui mène du modèle translation aux modèles plus complexes polynomiaux, en passant par l'affine et le projectif réduit.

B. Architecture et implémentation

1. Les grandes étapes

L'architecture globale du prototype se découpe en 3 grandes phases :

- appariement de points d'intérêt,
- vote et estimation de transformées d'image,
- et construction d'image.

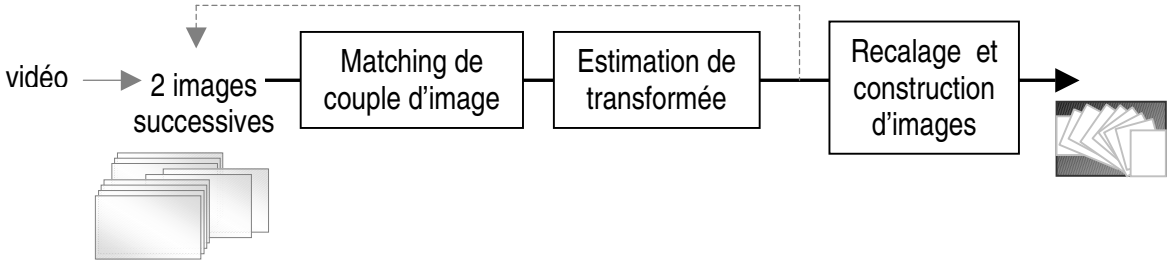


Figure 11 : les grandes étapes du mini-éditeur

En annexes, une description détaillée de chaque partie est proposée (p 64 à p66).

2. Appareillage de deux images successives

La première phase du développement concerne l'appariement de deux images afin de déterminer les déplacements des pixels d'une image à l'autre.

a) Structures implémentées

- Position.h/.c

Ces fichiers implémentent la gestion de tableau de positions avec des fonctions d'allocation mémoire, d'édition, de sortie, de recherche et de trie développés au fur et à mesure de l'avancement du projet.
La position élémentaire contient deux flottants, nous permettant ainsi de travailler sur une précision au sub-pixel près.

- VectorMotion.h/.c

Les structures sont similaires à celles de Position.h, et expriment un déplacement d'un pixel d'une image à l'autre.

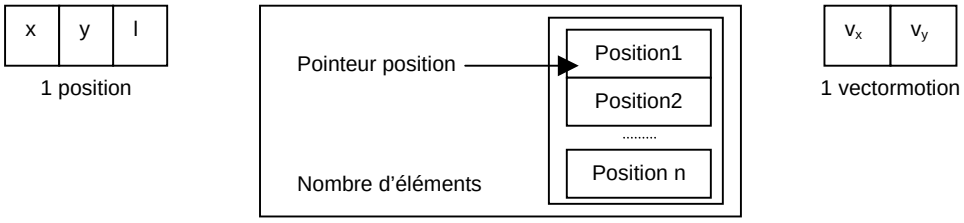


Figure 12 : structures de positions et de vecteurs mouvements

Remarque : il peut paraître inutile de créer toute une structure Position avec une allocation de pointeur, alors qu'il est possible de manipuler directement les tableaux. Ce choix personnel se justifie car il permet :

- de faciliter le passage des structures dans les fonctions en donnant seulement un pointeur, et satisfaisant une vue de l'esprit,
- de faire évoluer la structure par des ajouts de nouveaux membres comme des informations statistiques ou autres (très utile lors d'expérimentations).

- Vector.h/.c

Cette librairie a été développée par Olivier Buisson. Elle propose toute une batterie de fonctions pour utiliser des structures « vecteur » et « images de vecteurs ».
Plutôt que de créer une structure contenant un tableau de signatures, nous utiliserons des « images de signatures ».

- DistanceHeapMotion.c/.h

Cette librairie contient les structures et les fonctions permettant de stocker les informations d'appariement entre deux images.
Une structure « DistanceHeapMotion » est constituée :

- d'une position de référence de l'image,
- d'un tableau de k positions candidates, des «knn» (k nearest neighbours), interprétées comme pertinentes pour un appariement,
- des informations de distance, de poids, etc.

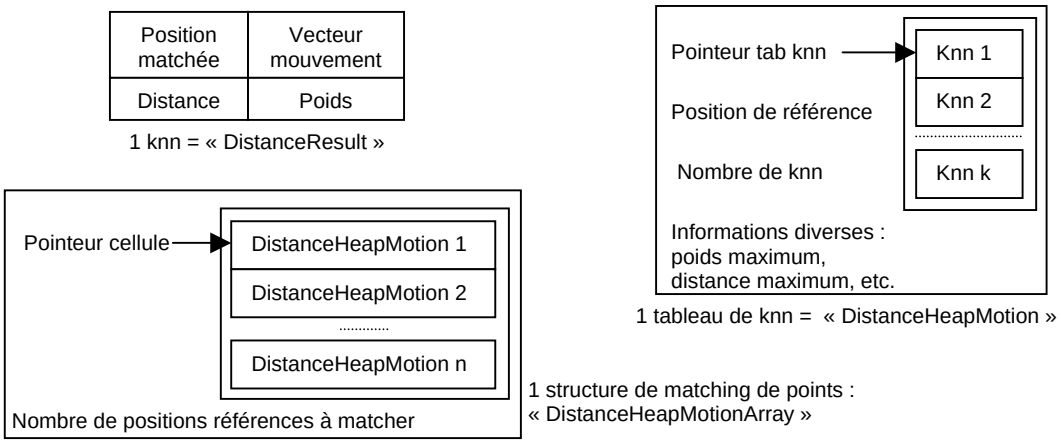


Figure 13 : structures de matching de points

- MatchingStructure.c/.h

Cette dernière structure englobe toutes les autres et contient les données en jeu lors de cet appariement d'images :

- un tableau de positions de références de l'image In,
- un tableau de positions sélectionnées dans l'image I_{n+1},
- deux images de descripteurs associées à I_n et I_{n+1},
- un tableau de correspondances de positions.

- DistancePositionStructure.h/.c

Cette librairie permet l'utilisation d'une structure intermédiaire pour trier les correspondances entre positions et ne retenir que les plus pertinentes, les knn.

b) Sélection de positions

(1) Indépendante du contenu de l'image

- fichiers « GridSelection.h/.c »

La fonction implémentée permet de sélectionner des positions de l'images sur une grille régulière définie par un pas d'espacement vertical et horizontal.
L'intérêt est de pouvoir envisager un appariement brut avec un maximum de positions.

(2) Guidée : détecteur de Harris

- fichiers « CornerSelection.h/.c »

Cette libraires reprend les fonctions développées par Alexis Joly de l'équipe TTA pour détecter des points d'intérêt. Les paramètres principaux sont l'écart type du filtre, la fenêtre d'analyse sur l'image.

Le détecteur est ajusté de façon à obtenir un compromis satisfaisant entre quantité et dispersion des coins. En effet, plus l'écart type du filtre est petit, plus les coins sont nombreux mais aussi moins caractéristiques de l'image et un plus grand nomre d'«outliers» risque d'apparaître.

Image 4 : comparaison de sélection de points d'intérêt avec le détecteur de Harris

c) Calcul « d'images » signatures

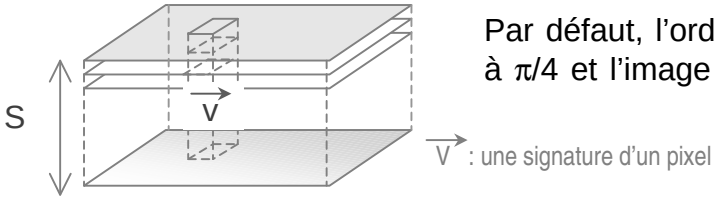
- fichiers « DistanceDetector.h/.c »
- fichiers « DensityDiscretCurve.h/.c »
- fichiers « ComputeImageDescriptor.h/.c »

Rappel : sur le principe une signature peut prendre la forme de tout jeu de données associé à une position. La signature est d'autant plus pertinente qu'elle permet de caractériser une position de manière indépendante aux autres pixels. Si nous parlons en terme de « distance », une signature est pertinente si elle éloignée de celle des autres pixels.

Le mini-éditeur emploie un filtre orienté paramétrable pour calculer un vecteur sur chacune des positions. Le principe est de calculer des dérivées autour d'un pixel selon différentes directions. La taille des vecteurs est donnée par :

S = (ordre maximum des dérivées) x (le nombre de rotations)

Il est possible de convertir l'image de vecteurs en ByteImage ; elle peut donc être considérée comme ayant une profondeur de Sx8 bits.



Par défaut, l'ordre est fixé à 2 et le pas de rotation à $\pi/4$ et l'image a donc une profondeur de 64 bits.

Figure 14 : image de « static vector »

Remarque : il est intéressant d'observer les différentes couches de l'image statique dans lesquelles on peut distinguer les dérivées spatiales orientées.

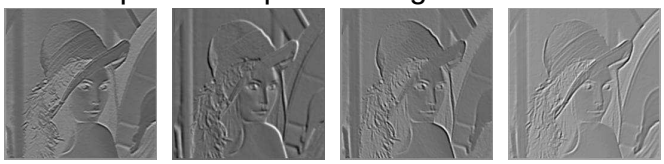


Image 5 : "static image vector" converties en ByteImage

Implémentation : la fonction computeImageDescriptor de haut niveau reprend les nombreuses librairies du projet signature pour calculer l'image « signature » d'une ByteImage.

d) Recherche de knn

- MatchPointWithDescriptor.h/.c

Lors d'une comparaison de données, il est préférable de considérer plusieurs candidats potentiels plutôt que de n'en garder qu'un. Un « knn » (k nearest neighbour) est un ensemble de données jugées similaires à une donnée de référence.
La fonction implémentée dans cette librairie recherche, pour chacune des positions de référence, un ensemble de candidats de l'image à « matcher » compris dans un cercle (le rayon de cette zone est entré dans la ligne de commande d'exécution du mini-éditeur).

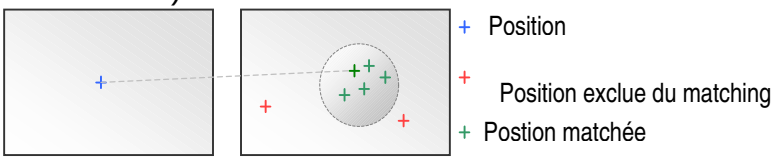


Figure 15 : "point matching"

(1) Algorithme

sélection d'un point de l'image référence
    sélection d'une position de l'image à matcher
        si cette position est comprise dans la zone de recherche
        alors calcul la distance L2 entre les descripteurs
            stockage dans la structure de trie « DistancePositionStructure »
        trier (q-sort) toutes les positions matchées par ordre croissant de distance dans l'espace des descripteurs
        copie les knn, c'est-à-dire les positions matchées détenant la plus faible distance, dans la structure de match « DistanceHeapMotion »

(2) Calcul de distances entre positions

- Vector.h/.c
- DistancePositionStructure.h/.c

Ces librairies contiennent respectivement les fonctions de calculs de distance entre descripteurs et une structure de trie de positions.

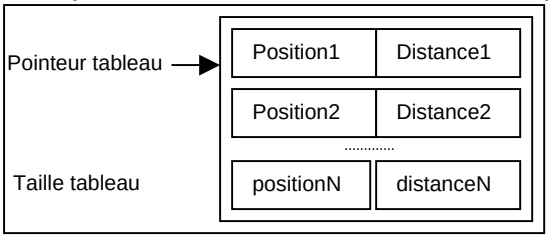


Figure 16 : la structure pour trier les « match »

La distance entre les vecteurs descripteurs est réalisée avec une simple distance euclidienne qui s'avère suffisante pour obtenir un panel assez large de valeurs.

Le tri utilise un algorithme rapide « q-tri » ou « quick sort » de la librairie standard sdtlio.h. Son implémentation s'effectue en deux étapes :

- une fonction de comparaison définissant le critère de comparaison,
- une fonction utilisant la fonction q-sort avec les bons arguments : un pointeur sur le tableau à trier, la taille du tableau, la taille de chaque élément du tableau, un pointeur sur la fonction de comparaison.

Remarque : le troisième argument ne nécessite pas la taille de l'élément à trier (en l'occurrence un flottant), mais la taille de la cellule contenant le flottant.

e) Attribution de poids

Immédiatement après l' appariement de positions, un premier poids ω_d est affecté de la manière suivante : plus la distance est petite, plus la correspondance de positions se verra attribuer un poids tendant vers 1.

$\omega_d = e^{-d/2\sigma}$

L'écart type donne la distribution du poids et s'ajuste par expérimentation.

Remarque : ω_r , un second poids est affecté par la suite, notamment lors de l'estimation de transformation (algorithme RANSAC) pour distinguer les correspondances «inliers» des «outliers».

$\omega_t = \omega_d * \omega_r$

L'appariement étant accompli, nous possédons toutes les informations nécessaires à l'estimation de mouvement.

3. Estimation de transformées d'images

Progressivement, ce sont trois 3 « circuits » que peuvent parcourir les données pour différentes estimations (automatiquement dirigés en interne selon les options mentionnées dans la ligne de commande).

L'ambition de mon développement est d'implémenter un modèle le plus général possible (ne correspondant à aucune transformée cohérente) :

$$x = \frac{tx + \frac{\cos\theta}{Sx} x' + \frac{\sin\theta}{Sx} y' + a_3 x' y' + a_4 x'^2 + a_5 x'^2 y' + a_6 x' y'^2 + a_7 y'^2 + a_8 x'^2 y'^2}{mx + ny + p}$$

$$y = \frac{ty - \frac{\sin\theta}{Sy} x' + \frac{\cos\theta}{Sy} y' + b_3 x' y' + b_4 x'^2 + b_5 x'^2 y' + b_6 x' y'^2 + b_7 y'^2 + b_8 x'^2 y'^2}{mx + ny + p}$$

a) Structure principale implémentée

- Homography.h/.c

Cette librairie contient les fonctions de gestion de structures « transformées d'images ». Une « homographie » possède principalement deux tableaux de 9 coefficients (le modèle est donc limité à un polynôme de degré 2), un tableau de 3 coefficient (pour le modèle projectif linéaire), une distance et un indice d' «inliers» permettant de sélectionner la meilleure transformée.

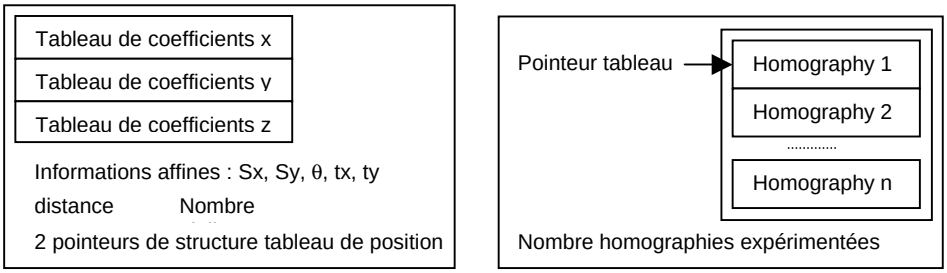


Figure 17 : 1 structures homographie et « tableau d'homographies »

b) Algorithmes RANSAC

- PseudoRansac.h/.c
- Ransac.h/.c

Selon [TZ 99], 40% des correspondances seulement sont valables si on utilise le détecteur de Harris. Ils proposent alors la méthode RANSAC pour une estimation optimale de transformées d'images. J'ai implémenté une librairie Ransac et une autre (« PseudoRansac ») reposant plutôt sur :

- une sélection soignée de correspondances de points,
- des conditions géométriques limitant les transformées d'images,
- un critère de distance géométrique entre images,
- de multiples tentatives d'estimation.

(1) Sélection pseudo-aléatoire de correspondances

- IndicePairMatchingArray.c/.h

Cette structure (et fonctions associées) permet de gérer une sélection pseudo-aléatoire de knn pour l'estimation des transformées. Ses fonctions permettent :

- de **filtrer** les correspondances pertinentes selon leur poids,
- d'éviter les positions redondantes ou non significatives (par exemple (0,0) appareillée avec (0,0)),
- d'imposer des règles géométriques entre correspondances.

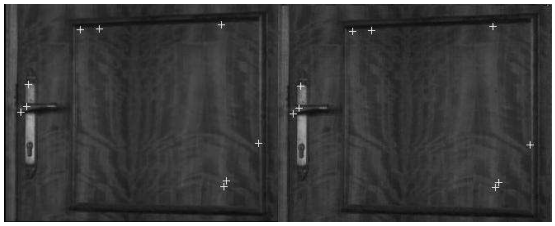


Image 6 : deux exemples sélection pseudo aléatoire de knn

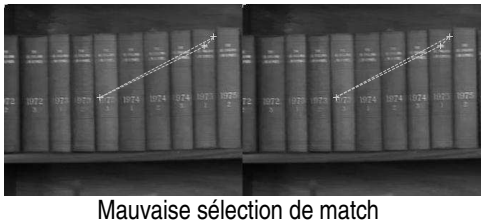
- **Sélection aléatoire de match**

Le générateur pseudo-aléatoire est initialisé une seule fois avec l'horloge système, par la fonction srand et donne une suite de nombres. A chaque appel, cette fonction renvoie alors un nouvel élément de cette suite.

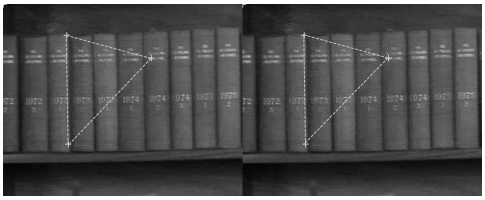
- **Contraintes de sélection de correspondances :**

Cette démarche anticipe l'appariement guidé mentionné dans l'algorithme de Torr et Zisserman [CZ]. Certaines fonctions de la librairie IndicePairMatchingArray imposent des règles géométriques pour augmenter la validité des transformations calculées. Par exemple, une fonction intervient notamment dans le cas affine pour éviter de sélectionner 3 positions trop proches et formant un angle trop fermé

.



Mauvaise sélection de match



Bonne sélection de match

Image 7 : sélection guidée de correspondances

(2) Calcul de transformées d'image

In **In+1**

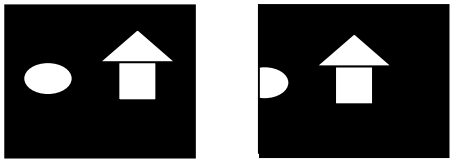


Image référence Image matchée

$t_x = x_{ref} - x_{match}$

$t_y = y_{ref} - y_{match}$

(a) Système linéaire

(i) Détermination du modèle translation

Pour estimer la translation, il suffit de prendre un seul couple de correspondances pour déduire le vecteur déplacement.

(ii) Détermination du modèle affine

Le modèle affine implique la résolution de 2 systèmes linéaires à 3 inconnues pour chacune des coordonnées.

Après la sélection de 3 correspondances de positions, nous pouvons établir deux jeux de trois équations exprimant le passage de l'image In+1 à sa précédente In:

$X_0 = a*X_0' + b*Y_0' + c$ et $Y_0 = d*X_0' + e*Y_0' + f$

$X_1 = a*X_1' + b*Y_1' + c$ $Y_1 = d*X_1' + e*Y_2' + f$

$X_2 = a*X_2' + b*Y_2' + c$ $Y_2 = d*X_2' + e*Y_2' + f$

Avec a, b, c, d, e, f les coefficients à rechercher,
(X_i, Y_i) la position de l'image I_n « matchée » avec (X_i', Y_i') dans l'image I_{n+1}.

La résolution de ces systèmes linéaires s'effectue avec des fonctions de la bibliothèque GSL sous forme matricielle.

(b) Estimation du modèle de projection linéaire réduit à 4 paramètres

Son estimation modélise le panoramique. La caméra effectue uniquement un mouvement de panoramique autour de son axe vertical, sans zoom et sans rotation autour du centre de l'image (sans instabilité). Techniquement, ce type de plan doit être tourné sur pied.
Le modèle homographique à 8 paramètres se trouve donc réduit à 4 paramètres sous ces conditions.

$$\begin{bmatrix} \frac{\cos\theta}{Sx} & \frac{\sin\theta}{Sx} & tx \\ -\frac{\sin\theta}{Sy} & \frac{\cos\theta}{Sy} & ty \\ m6 & m7 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ devient } \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ m6 & m7 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Il existe une méthode simple pour estimer ce modèle, citée dans [HZ 00] que je n'ai pu me procurer. Une fois cette fonction implémentée, le modèle sera immédiatement fonctionnel, sans développement complémentaire : mise sous forme matricielle, elle peut être reprise par le circuit « affine ».

(c) Systèmes non linéaires : modèle polynomial

Si l'on complexifie l'équation affine le système n'est plus linéaire. Pour chaque jeu de correspondances, il n'existe pas une unique transformée mais une infinité. Si les propriétés linéaires de la transformation affine permettent de déterminer les coefficients de son équation à partir d'un jeu de 3 correspondances seulement, les transformées plus complexes, quant à elles, impliquent une estimation de la transformée optimale.

Il faut « observer » N couples de correspondances de points afin d'estimer une équation liant au mieux les positions appareillées. La technique de l'**ajustement des moindres carrés** permet de trouver une équation rapidement. Le principe est d'estimer l'équation optimale d'une courbe passant au plus près des points du jeu de données, en minimisant un critère de distance euclidienne.

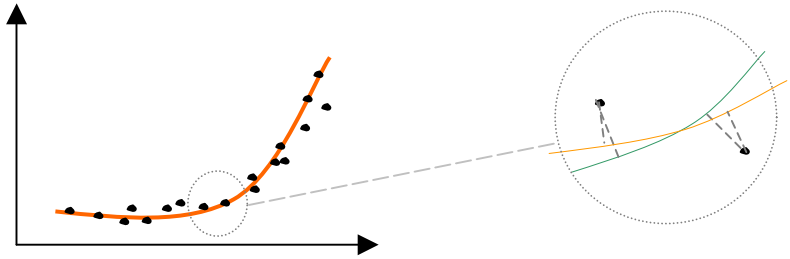


Figure 18 : ajustement des moindres carrés

Le modèle polynomiale implique deux estimations d'équations :

$$x=\sum_{i=0}^{n}\sum_{j=0}^{m}C_{ij}x^i y^j \qquad y=\sum_{i=0}^{n}\sum_{j=0}^{m}D_{ij}x^i y^j$$

Les observations sont les correspondances de positions entre les deux images, Cij et Dij les coefficients à estimer.

Remarque1 : le nombre d'observations doit être tout de même limité car il risque d'augmenter la présence de correspondances «outliers».

Remarque2 : il est possible avec cette méthode d'estimer un modèle linéaire.

(3) Critère de sélection de transformée

(a) Première version basée sur la distance géométrique d'images

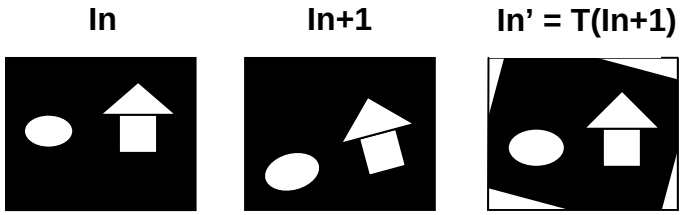


Figure 19 : transformation d'images

- ImageHomographyFunction.h/.c

Un critère simple de sélection de transformée repose sur une distance géométrique entre image. Les ByteImages de Dali sont des vecteurs de « caractères », (compris entre 0 et 255) et de taille longueur*hauteur.
La distance alors associée à l'homographie calculée correspond à une distance euclidienne entre le vecteur image de référence I_n et l'image I_n' matchée et transformée. Plus T fait tendre I_{n+1} vers I_n, et plus la distance est petite. La transformée retenue minimise cette distance.

Remarque : il faut initialiser l'image I_n' par la moyenne en luminance de I_n pour être plus précis puisque I_n' doit tendre vers I_n.

(b) Deuxième version basée sur la distance de coins

- ImageHomogrphyFunction.h/.c

Théoriquement, si T est parfait $I_n'=I_n$. Par conséquent les positions des coins sont identiques entre les deux images.
Le critère de pertinence de la transformée peut donc être la distance entre descripteurs à ses positions « coins ».

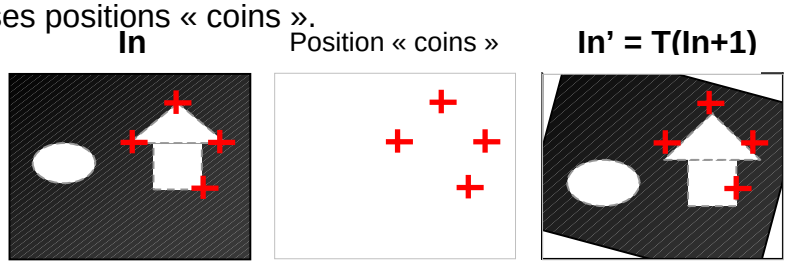


Figure 20 : critère de sélection d'homographie par recensement d'«inliers»

Cette fois-ci plutôt que retenir les distances totales entre descripteurs, la fonction implémentée compte le nombre de correspondances «inliers». Si la distance entre descripteurs est inférieure à un seuil, cette position est alors considérée comme un «inlier » et dans le cas inverse comme un «outlier». La transformée retenue est celle qui maximise le nombre d'»inliers».

(4) En option : limitation des transformées

- Homography.h/.c

Cette fonction intervient si l'estimation robuste n'est pas activée. Historiquement, l'estimation optimale ne fut implémentée que tardivement, et la plupart des tests illustrés dans ce rapport ont été réalisés à partir d'homographies non optimisées (ce qui ne signifie pas que les résultat sont mauvais…).

Le principe de l'algorithme de « PseudoRansac » est de calculer un grand nombre de transformées pour obtenir la meilleure. Une boucle « tant que » attend qu'un nombre donnée de transformées soit calculé pour choisir ensuite la meilleure. Deux fonctions viennent limiter les coefficients des transformées afin que le choix s'effectue sur un plus grand nombre de transformations pertinentes.

Des conditions limites subjectives sont définies au pré-processeur en macro dans le fichiers homograpy.h. La premier coefficient, celui correspondant grossièrement à la translation, est limité à une valeur liée à la résolution de la vidéo. Dans le cas d'un transformation affine les facteurs d'échelles et l'angle de rotation sont limités dans une certaine plage de valeurs définie par :

$$x = ax'+by'+c$$
$$y = dx'+ey'+f$$

avec

$$\frac{1}{S_{\min}} \le a \le \frac{\cos\theta_{\max}}{S_{\max}} \quad idem\ pour\ e$$
$$-\frac{\sin\theta_{\max}}{S_{\max}} \le b \le \frac{\sin\theta_{\max}}{S_{\min}}\ idem\ pour\ (-d)$$
$$t_{\min} \le t_x \le t_{\max} \quad et \quad t_{\min} \le t_y \le t_{\max}$$

Très subjectivement, nous pouvons considérer qu'il est rare d'avoir deux images successives avec un mouvement global de plus de θ_{max} =10° et un facteur d'échelle de plus de $S_{max} = 1/S_{min} = 1,5$.

c) Raffinement d'homographie : algorithme Levenberg-Marquardt

- NonlinearEstimation.h/.c

J'ai implémenté cet algorithme sans arriver à obtenir des résultats satisfaisant. Toutefois, je tente une explication qui sera utile pour la poursuite du projet.
Le principe est de faire converger, à partir des positions «inliers» trouvées par Ransac le modèle préestimé vers un modèle optimal. J'ai repris l'exemple proposé dans la documentation de GSL pour implémenter cette fonctionnalité. L'implémentation se divise en 4 points principaux :

- définition du modèle,
- implémentation de la matrice Jacobienne,
- initialisation du modèle,
- itération et convergence du modèle.

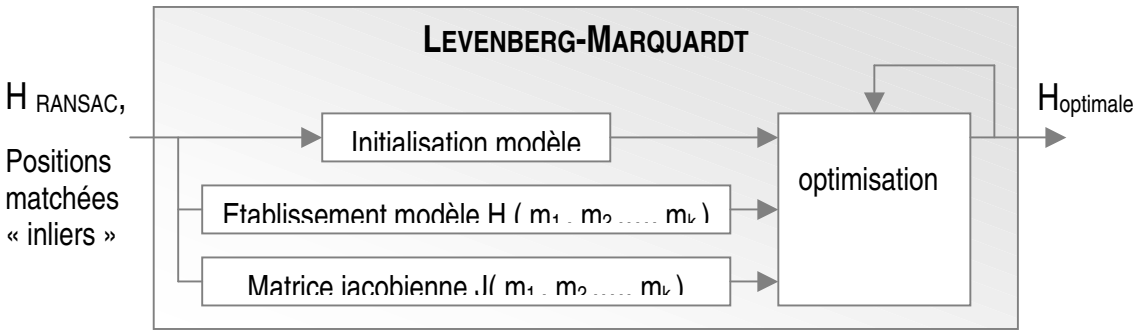


Figure 21 : étapes de l'algorithme Levenberg Marquardt

- Définition du modèle :

Soit un couple de match (xi, yi et xi', yi') lié par une expression $g(xi', yi')$ en fonction de paramètres à estimer : m_0, m_1, …
Le critère à minimiser est un vecteur $Xi = f((xi, yi), (xi', yi'))$

- matrice Jacobienne

La matrice Jacobienne doit avoir pour taille KxN, avec K le nombre de paramètres du modèle à optimiser et N le nombre d'»inliers» à partir duquel il est calculé. Les éléments de la matrice Jacobienne reprennent les dérivées partielles de l'expression :

$$J = df/dp = \begin{bmatrix} \frac{\partial f(x_0, y_o)}{\partial m_0} & \frac{\partial f(x_0, y_o)}{\partial m_1} & \frac{\partial f(x_0, y_o)}{\partial m_2} & \cdots & \frac{\partial f(x_0, y_o)}{\partial m_k} \\ \frac{\partial f(x_1, y_1)}{\partial m_0} & \frac{\partial f(x_1, y_1)}{\partial m_1} & \frac{\partial f(x_1, y_1)}{\partial m_2} & \cdots & \frac{\partial f(x_1, y_1)}{\partial m_k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f(x_m, y_m)}{\partial m_0} & \frac{\partial f(x_m, y_m)}{\partial m_1} & \frac{\partial f(x_m, y_m)}{\partial m_2} & \cdots & \frac{\partial f(x_m, y_m)}{\partial m_k} \end{bmatrix}$$

La fonction de raffinement du modèle demande à ce qu'on initialise les paramètres à optimiser : il suffit alors de lui passer les coefficients de la transformée déterminée par l'algorithme Ransac précédent. Ensuite le système construit le modèle et sa Jacobienne et lance une boucle itérative pour faire converger les paramètres vers un modèle optimal.

- Difficulté :

La principal difficulté consiste à bien choisir l'expression de son modèle. Par exemple, je souhaite estimer les coefficients de la projection linéaire. J'ai donc 9 coefficients à regrouper sous une même expression, car les deux équations utilisent les mêmes paramètres :

$x = \frac{ax'+by'+c}{mx'+ny'+p}$

$y = \frac{dx'+ey'+f}{mx'+ny'+p}$ que je peux regrouper par exemple dans $x + y = \frac{ax'+by'+c+dx'+ey'+f}{mx'+ny'+p}$ ce qui est incomplet mathématiquement parlant.

Il vaut mieux alors implémenter l'expression en distance géométrique : $x^2 + y^2 = \frac{(ax'+by'+c)^2 + (dx'+ey'+f)^2}{(mx'+ny'+p)^2}$ ce qui alourdi la matrice Jacobienne.

4. Recalage d'images

Problématique : la boucle principale du programme effectue des appariements d'images par deux. La transformée d'image déterminée exprime le passage de la seconde vers la première dans un repère associé. Au final, nous obtenons une tableau de transformées associées à chaque couple d'image.

Si l'on veut pouvoir recaler toutes les images ensembles, il faut déterminer un repère commun et absolu pour réévaluer et lier les transformées.

Image 8 : David Hockney

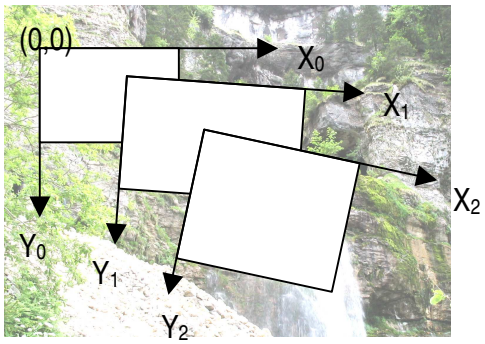
a) Structure implémentée

- IndicePairMatchingArray.c/.h

Cette structure recueille toutes les transformées calculées sur la séquence et possède des fonctions d'analyse et de réévaluation des homographies pour déterminer la taille finale de l'image mosaïque et les positions relatives des images.

b) Cas affine

(1) Combinaison matricielle



Le mini-éditeur appareille des couples d'images pour estimer une transformée géométrique. Cette transformée exprime alors les coordonnées de la deuxième image dans le repère de la première.

Figure 22 : position relative des images recalées dans un repère absolu

Le mode affine permet de représenter ces transformées sous forme matricielle et il est alors aisé de recaler les images les unes par rapport aux autres. Il faut fixer une image, par exemple la première, comme étant celle qui porte le repère absolu et sa matrice associée devient la matrice identité. Chaque matrice de transformation doit alors être recalculée dans ce nouveau repère en la multipliant avec chacune des matrices qui la sépare de la matrice identité.

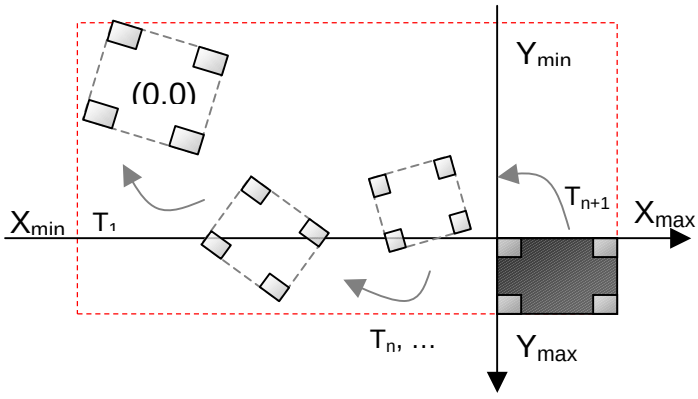
En effet, la matrice H_n induisant le passage de l'image I_{n+1} vers I_n devient :
$H_n' = H_{n-1} * H_n$ pour exprimer le passage de I_{n+1} vers I_{n-1}, puis $H_n'' = H_{n-2} * H_{n-1} * H_n$ exprime les coordonnées de I_{n+1} dans le repère de I_{n-2}, etc.
Toute transformation affine recalée est trouvée par :

$$H_{n\ recalé} = \prod_{i=n}^{0} Hi$$

Rappel : la multiplication de deux matrices ne donne pas le même résultat si elles sont multipliées dans un sens ou dans l'autre. Il faut bien « remonter » jusqu'à la matrice identité.

(2) Détermination de la taille finale

Une fois les transformées ramenées dans un même repère, il est simple de déterminer la taille finale de l'image fusion. Etant donnée que le modèle affine est une transformation rigide, les positions des 4 coins de l'image resteront les 4 coins de l'image transformée.
Une fonction sélectionne les 4 positions coins de la dernière image de la séquence et passe à travers chaque transformation. A chaque itération, les coordonnées extrêmes sont retenues si elles dépassent, en valeur absolue, les précédentes. A la fin du processus, les coordonnées extrêmes sont connues et permettent de déterminer la taille finale de l'image mosaïque et un « offset », soit une translation, pour recentrer toutes les images en coordonnées positives.



$L_{mosaïque} = X_{max} - X_{min}$
$H_{mosaïque} = Y_{max} - Y_{min}$

$tx_{offset} = - X_{min}$
$ty_{offset} = - Y_{min}$

Figure 23 : détermination de la taille finale de l'image mosaïque, cas affine

c) Modèles polynomiaux

Avec des modèles plus complexes, dont les équations sont non linéaires, il devient impossible de recalculer les transformées pour un recalage. Dans un premier temps il faut déterminer la taille finale de l'image mosaïque, puis travailler sur chacun des pixels.

(1) Taille d'une image mosaïque d'un modèle polynomial

Le processus est quasi similaire au cas affine mise à part que la recherche des coordonnées extrêmes s'effectue non plus sur les coins de l'image mais sur l'ensemble des positions de l'image. En effet, ces transformations ne sont pas rigides et une position située au centre de l'image peut devenir une position du bord de l'image transformée et recalée.

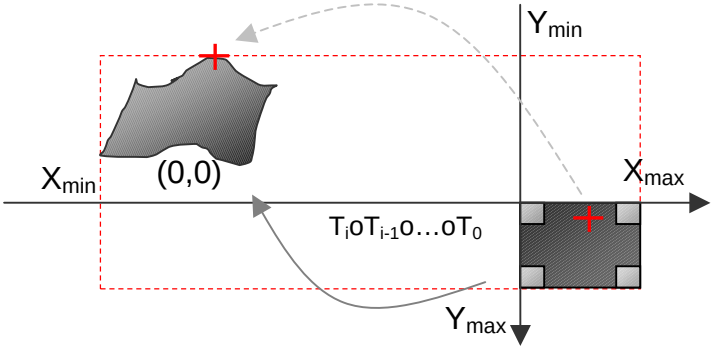


Figure 24 : détermination de la taille finale de l'image mosaïque, cas générique

(2) Recalage des images

Etant donné qu'il est trop lourd (mathématiquement parlant) de recalculer les transformées dans le repère absolu, ce ne sont plus les images qui vont être recalé mais les pixels.

L'algorithme mis en place suit un processus itératif :

1. sélection de l'image I_i à recaler de 0 à n
   a. sélection d'une position j dans cette image
      i. passage de cette position à travers chacune des transformées précédentes
      ii. arrondi de cette position

Limite :
L'image mosaïque est une ByteImage avec des positions en entier. Or nous travaillons sur des positions en flottant. En conséquence, l'image finale n'est pas assez précise avec un simple arrondi. L'idéal serait de pouvoir faire des interpolations, mais une telle fonction semble assez lourde à réaliser. Je suggère de récupérer une fonction de transformation polynomiale d'une bibliothèque qui sera certainement optimisée.

Remarque : la bibliothèque Dali ne propose pas de telle transformée. En revanche, les éditeurs d'images comme « The Gimp » possèdent des fonctions de distorsion qui peuvent être récupérées dans un projet.

V. Résultat visuel et analyse

Au-delà de la simple construction d'images mosaïques, le sujet reste assez ouvert et offre des possibilités de création originale. La première application, la plus simple, est « l' addition » des images, comme un photomontage.

A. Préliminaires

1. Remarques sur l'emploi du mini-éditeur

1. Le mode « corner » semble être suffisant pour matcher deux images et permet un gain de temps de calcul par rapport au mode grille.
2. Le pas de match entre images de séquences doit être de préférence réglé à 1, sauf dans le cas de long zoom lent. En effet, beaucoup de travellings optiques impliquent une migration des pixels de l'ordre de l'unité et fausse à la longue l'estimation des transformées.
3. Il est préférable de limiter le jeu de correspondances pour estimer une transformée et de parier sur de nombreuses tentatives (100 à 500 pour les modèles affines, 1000 à 3000 pour les modèles polynomiaux).

2. Rappel sur les notions d'angle et de profondeur de champs

Avant de pouvoir faire une interprétation des images réalisées, il semble important de rappeler quelques notions de prise de vue pour pouvoir mieux justifier les observations.

Lors d'une prise de vue la profondeur de champ dépend à la fois de la focale, du diaphragme, et de la distance entre l'objectif et le sujet cadré, c'est-à-dire la distance de mise au point. Ainsi,

- plus le diaphragme est fermé, plus le nombre d'ouverture est grand et plus la profondeur de champ est grande,
- plus la focale de l'objectif est longue, plus la profondeur de champs est faible,
- plus l'objet cadré est éloigné de l'objectif, et plus la profondeur de champs est grande.

La notion d'angle de champ dépend de la focale induite par le choix de l'objectif. Un objectif à focale courte ou « **grand angle** » donne des plans larges à faible distance. La perspective est accentuée et « accélère » les déplacements des objets dans l'axe de l'objectif. Les **téléobjectifs** à focales longues couvrent un angle de champ étroit et donne des plans très serrés à de grandes distances. Les fuyantes sont peu marquées, la perspective est écrasée et la profondeur de champ est faible.

Remarques : le grand angle est donc plus adapté pour la prise de vue de panoramique et de travelling (s'il n'y a pas de poursuite d'objet avec variation de focale pour garder la mise au point).

B. Validation et limites des modèles

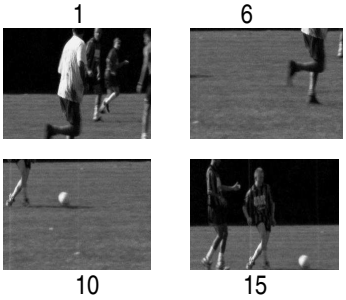
J'ai développé des petits éditeurs de séquences d'animation à partir d'une image fixe. Les séquences ainsi réalisées m'ont servi pour créer plusieurs classes de mouvements de caméra et pour établir une vérité « terrain » utile lors des appariements des points d'intérêts. Ces petits logiciels permettent :

- de créer des mouvements rigides d'objets dans une image (translation, rotation, échelle),
- de « filmer », de recadrer dans une grande image ou dans une séquence vidéo.

1. Modèle translation

Le modèle « translation » fonctionne particulièrement bien pour un mouvement de travelling et de panoramique, mais n'est plus valable pour un zoom. La plupart des travellings et panoramiques impliquent une migration plutôt lente des pixels, d'une image à l'autre. La superposition des images implique alors un ajout par tranches « fines ».

- Validation du modèle :



Le modèle est testé par une séquence dans laquelle une «caméra » cadre une image fixe progressivement dans toutes les directions en suivant un mouvement circulaire.

Image 9 : modèle translation, validation omnidirectionnel

- Limites du modèle :

L'inconvénient majeur de cette méthode est de provoquer d'importantes déformations de la scène à cause de deux facteurs :

- la réduction de mouvements circulaires, notamment lors de panoramique, à un mouvement latéral,
- l'ajout par tranche assemble les zones de l'image les plus déformées par le système optique de la caméra, les bords, comme vous pouvez le constater dans cet exemple :

Image 10 : mouvement panoramique modélisé par une translation sur la séquence « Afrique »

De plus, une jonction apparaît nettement (ici entourée) à l'ajout de la deuxième image. Dans cet exemple, l'ajout se fait « par-dessous » : la première image de la séquence se retrouve à gauche et les suivantes s'ajoutent comme si elles étaient dessous.

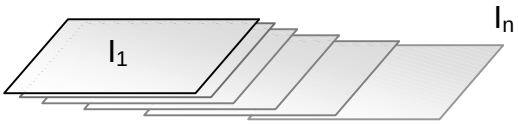


Figure 25 : ajout de "tranches" d'images par-dessous

Remarque 1: j'ai implémenté le mini-éditeur de manière à pouvoir reprendre la séquence d'image de manière anti-chronologique, et donc de pouvoir reconstruire la séquence d'images par « dessus ».

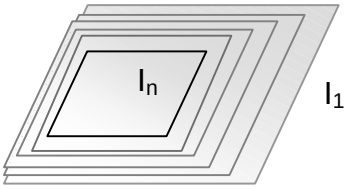


Figure 26 : ajout de tranche par-dessus lors d'un zoom entrant

Remarque 2 : cette possibilité s'avéra indispensable lors d'un zoom entrant, sinon le résultat final ne correspond qu'à la première image qui obstrue toutes les autres.

Deux propositions pour atténuer ces déformations :

- perfectionner le modèle vers un modèle plus complexe,
- ajouter les images entre elles non plus par tranches mais par cercles superposés pour limiter la déformation de la lentille sur les bords de l'image.

2. Modèle affine

Le second niveau de complexification du modèle est l'introduction de la rotation et du changement d'échelle entre images. Le modèle affine semble pouvoir répondre à tout mouvement rigide du cadre.

a) Rotation

Pour valider ce mouvement de caméra, j'ai réalisé des séquences de rotation rigide d'objet, avec un changement de luminance du fond pour observer la superposition.

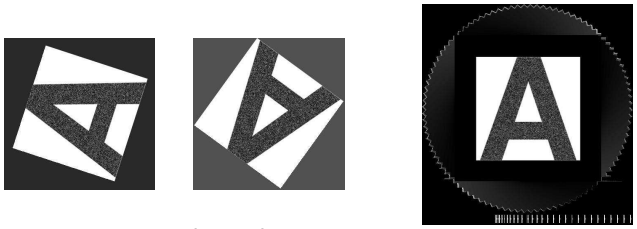


Image 11 : mouvement de rotation

- Limites du modèle :

Mais comme nous l'avertit Torr et Zisserman dans [TZ 99], la rotation peut être interprétée différemment. A priori, nous pensons pouvoir modéliser un travelling avec des mouvements type « steadycam », ou alors compenser les instabilités d'une captation caméra-épaule. Le problème est que la profondeur de champs fausse la modélisation de la rotation.

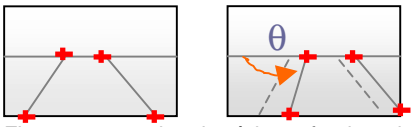


Figure 27 : rotation due à la profondeur de champs

Cet exemple reprend la même séquence que pour le modèle affine avec une compensation de déformation de la lentille. Nous pouvons observer l'effet de rotation du au changement de profondeur : le début de la séquence vidéo est constitué de deux plans, des bâtiments proches au premier plan, et le reste de la ville au second plan. Une légère rotation s'effectue à cet endroit alors que dans la deuxième partie, ne contenant pas de plan proche, aucun mouvement de rotation n'est perceptible.

Image 12 : modèle affine sur la séquence « Afrique »

b) Changement d'échelle

- Validation :

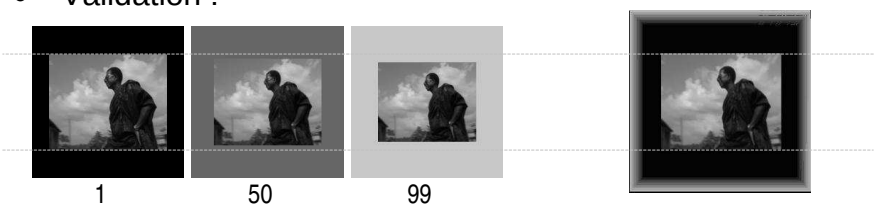


Image 13 : ajout par en dessous d'une séquence à changement d'échelle

- **Limites :**

Avec cette expérimentation (ci-dessous), nous pouvons voir que la mise à plat du travelling optique affecte peu, au premier abord, le résultat final. Il est intéressant de constater que l'image résultante est déformée vers l'extérieur alors que l'image de départ zoom incurve les lignes.

- **Supposition :**

Cette observation n'est peut être pas si anodine et fait écho au défaut de distorsion des lentilles d'un objectif : une focale courte possède la propriété de déformer les images en tonneau et une longue focale en coussinet. Or le zoom entrant passe, en exagérant, d'une position « grand angle » vers une position « téléobjectif ». Ainsi, il pourrait « imprégner » la mosaïque finale de ce changement de focale.

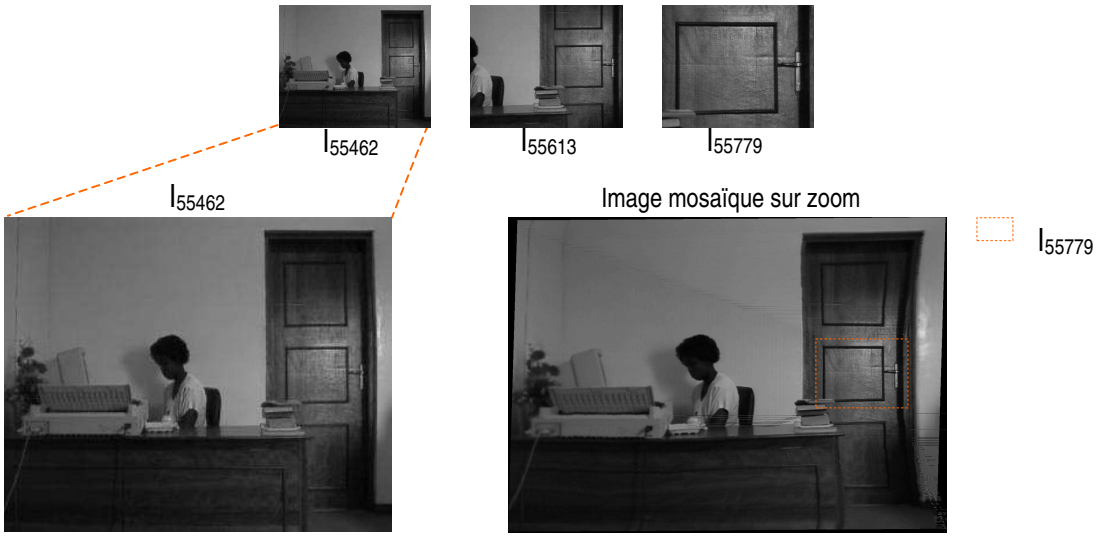


Image 14 : modèle affine sur un zoom

Malgré tout, cet exemple nous interpelle sur la faible différence entre l'image de départ du mouvement du cadre : dans le cas d'un zoom « pur », qu'est-ce qu'une image mosaïque apporte de plus à l'interprétation d'une scène ? Est-ce la déformation autour de l'objet zoomé induit une interprétation du mouvement de la caméra ? Devons nous dégrader l'image mosaïque, en créant des écart artificiels, ou ajouter des informations supplémentaires pour rendre cette interprétation plus lisible ?

c) Combinaison rotation – changement d'échelle - rotation

Nous pouvons finalement tester tous les mouvements affines combinés :

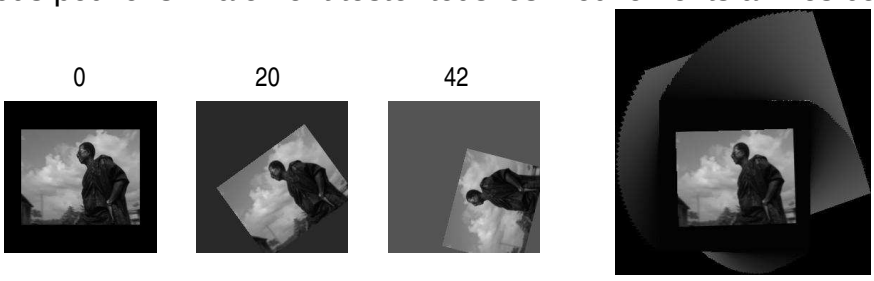


Image 15 : test de validation d'un mouvement affine

Image 16 : vue aérienne par modèle affine

3. Modèles polynomiaux

a) Modèle pseudo-perspectif

Comme nous l'avons vu dans l'état de l'art, le modèle de caméra perspectif implique une estimation de transformée « projection linéaire ».

$$x = \frac{ax'+by'+c}{mx'+ny'+p} \qquad y = \frac{dx'+ey'+f}{mx'+ny'+p}$$

Par expérimentation, il s'avère que ce modèle peut être réduit à un modèle polynomial complet au premier ordre :

$$x = a_0 + a_1 x' + a_2 y' + a_3 x' y'$$

$$y = b_0 + b_1 x' + b_2 y' + b_3 x' y'$$

Image 17 : panoramique selon un modèle polynomial au premier ordre

Dans cette image, les traits noirs sont dus à l'arrondi sur les positions flottantes.
Il est difficile de juger de la pertinence du modèle de caméra perspectif.

Toutefois, comme le rappelle Capel et Zisserman ([CZ]), l'objectif grand angle utilisé particulièrement dans le cas du panoramique induit une grande profondeur de champ. Cet effet est encore plus accentué si l'objet focalisé est éloigné de la caméra. En conséquence, un modèle de caméra projectif, utilisant des homographies complètes réintroduit toute cette perspective dans l'image mosaïque. Ainsi un panoramique va voir son centre fortement aminci, alors que le modèle de caméra orthographique formera une image rectangulaire mais en incurvant les lignes droites.

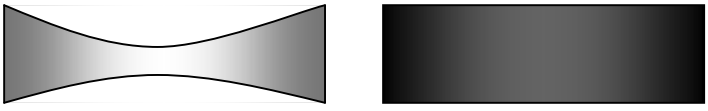


Figure 28 : mise à plat d'un panoramique avec modèle de caméra perspectif et orthographique

b) Modèle quadratique

C'est a priori le modèle le plus complet, dans le cas de séquences audiovisuelles, pour un modèle de caméra orthographique. Tous les degrés de liberté sont présents (translation, rotation, zoom).

$$x = a_0 + a_1 x' + a_2 y' + a_3 x' y' + a_4 x'^2 + a_5 y'^2$$

$$y = b_0 + b_1 x' + b_2 y' + b_3 x' y' + b_4 x'^2 + b_5 y'^2$$

Autant son estimation fonctionne bien avec un ajustement des moindres carrés (IV.B.3.b)(2)(c)), autant il est difficile d'obtenir des résultats visuels satisfaisants précis sauf dans le cas du zoom ou changement d'échelle pur comme nous l'avertissait déjà [SSO] (cf III.A.3.b)(1)).

- Validation des mouvements élémentaires :

La translation ne pose pas de problèmes particulier. J'ai utilisé le même test omnidirectionnel que pour la validation du modèle translation.
Le cas du changement d'échelle fonctionne bien comme l'illustre ces exemples. Je compare plusieurs images correspondant au nombre d'estimations d'homographies attendu dans RANSAC.

Figure 29 : cas du chngement d'échelle par modèle quadratique

Nous voyons naturellement que plus le nombre d'essais est important et plus le recalage devient précis. Idéalement, pour un nombre infini de tentatives, l'image recalée doit correspondre à la même que celle obtenue par le modèle affine.

Remarque : les erreurs d'arrondis sur les positions flottantes impliquent des pixels vides. Mais à chaque recalage d'images, ces positions sont remplies par l'accumulation des transformées (voir IV.B.4.c)(2).

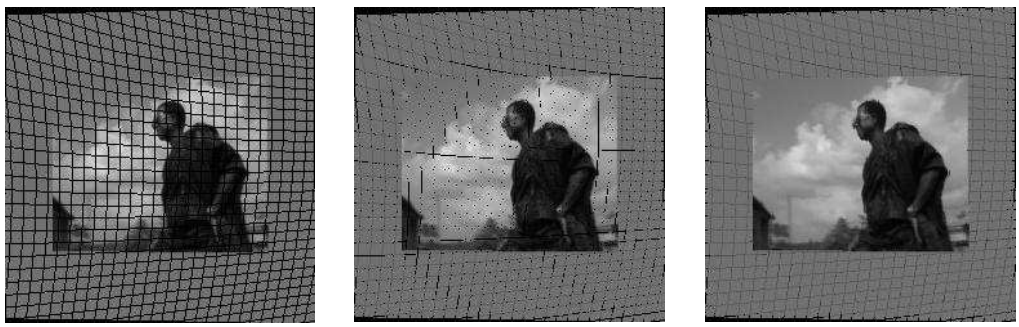


Image 18 : reconstruction d'image mosaïque par modèle polynomial

En revanche, mes tests n'ont pas permis de valider la rotation : les composantes en degré 2 viennent perturber la reconstruction.

Conclusion :
Il est difficile d'appliquer ce modèle si un mouvement de rotation est perceptible dans la scène. Ce modèle pourra être optimisé par l'emploi de l'algorithme de Levenberg-Marquardt par exemple.

c) Modèle polynomial du second ordre

$$x = \sum_{i=0}^{n} c_n x'^n \; y'^n \qquad y = \sum_{i=0}^{n} d_n x'^n \; y'^n \text{ avec n=2}$$

Le modèle complet du second ordre est difficilement contrôlable, et le système doit être perfectionné pour devenir plus robuste. Mais il reste à mesurer l'impact réel des composantes en xy^2, x^2y et x^2y^2.

4. Modèle projection linéaire réduit

Ce modèle est réduit pour le cas d'un panoramique avec 4 degrés de liberté. Je n'ai pu l'implémenter car il est difficile de déterminer les coordonnées homogènes des points. Plusieurs tentatives avec l'algorithme de Levenberg-Marquardt ont été vaines.

C. Comparaison de modèles

1. Translation vs affine

La séquence mise à plat ci-dessous est un travelling de gauche à droite finissant sur un léger zoom et sa mise au point. Le modèle translation donne a priori un résultat plus esthétique mais ne rend pas compte de l'effet de zoom final. Le modèle affine redonne une impression de profondeur en fin de séquence malgré de légères déformations.

Image 19 : travelling « bibliothèque » mis à plat par le modèle translation

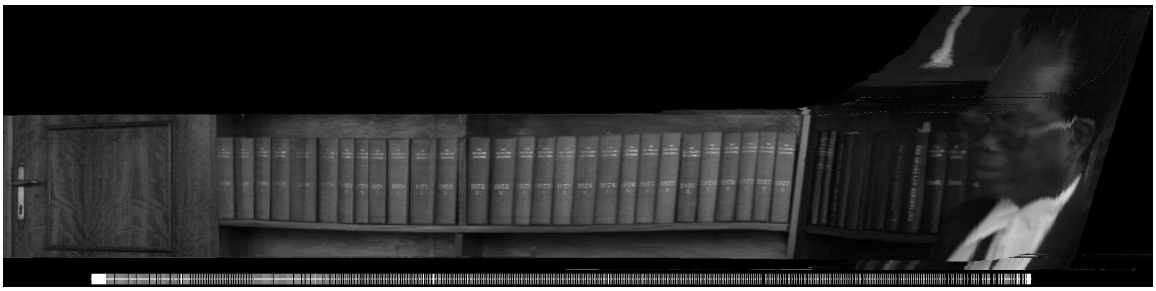


Image 20 : travelling « bibliothèque »mis à plat par le modèle affine

Les déformations sont notamment dues à des transformations calculées à partir de positions accrochées sur le personnage qui effectue un mouvement local.

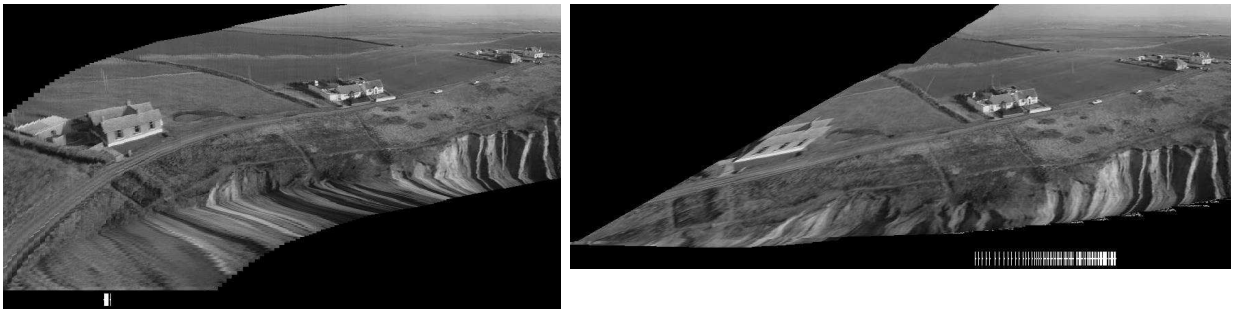


Figure 30 : mise à plat d'une prise de vue aérienne (translation et affine)

Dans cet exemple, la mosaïque « translation » paraît à première vue plus équilibrée. Nous y retrouvons les propositions rectangulaires de la vidéo. Mais les déformations sont flagrantes : les falaises sont complètement étirées contrairement à l'image créée par le modèle affine. La fin de la scène se termine sur une combinaison d'une rotation et d'un zoom sur les voitures. Seul le modèle affine rend compte de ce resserrement en finissant par une « pointe ».

Ces exemples montrent bien une des problématiques de visualisation de document audiovisuel liée à la distinction entre représentation spatio-temporelle et mise à plat : que décryptons-nous le mieux ? Une représentation préservant les proportions rectangulaires du cadre, ou une recomposition fidèle de la scène privilégiant l'« action » ?

D. Cas limites

1. Problèmes d'occlusion

Dans cette séquence prise à partir d'un hélicoptère, la caméra poursuit les véhicules en contournant la colline. L'image mosaïque est assemblée « par dessous » et ne dévoile donc que la première partie de la scène.

En revanche cette représentation rend compte du volume de la scène : la mosaïque reconstitue une partie de la butte.

Image 21 : image mosaïque dans le cas d'un changement d'axe

Ce cas de changement d'axe de la caméra nous permet d'envisager des images mosaïques projetées sur un modèle 3D pour palier au problème d'occlusion.

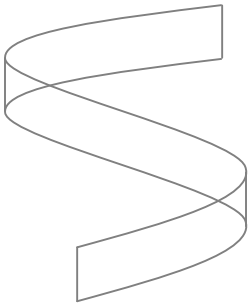


Figure 31 : idée de mise à plat en 3 dimensions

2. Cas extrême

Certaines prises de vues accumulent les difficultés pour estimer les homographies. Par exemple (ci-dessous), la scène est prise de l'intérieur d'une voiture impliquant des déformations optiques du par-brise. De plus, les coins ont tendance à être accrochés sur les objets du premier plan, ici lisse, donc avec peu de points d'intérêt indépendants. En conséquence, des coins de la voiture dans le fond, risquent d'être plus facilement employés pour estimer l'homographie. Or, cet objet est en mouvement et perturbe la précision de la transformée.

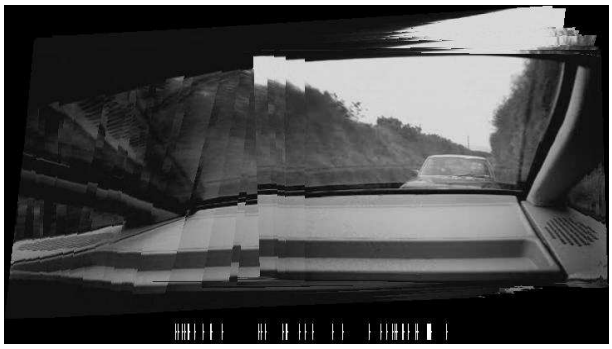


Figure 32 : problème de présence d'objet volumineux et déformation optique

La théorie sur l'estimation de mouvement doit pouvoir proposer des solutions pour « filtrer » ces mouvements locaux du mouvement global de la caméra.

3. Mini bilan

D'après ces observations et les recherches de l'état de l'art, nous pouvons établir un aperçu des transformations adéquates à différentes prises de vue.

Modèle	translation	Affine	Pseudo-perspectif	Perspectif_réduit	quadratique
Panoramique	Ok	Ok	Ok sous réserve d'interpolation	Lourd	Lourd
Travelling proche	Ok	La meilleure	Inutile	Inutile	Inutile
Travelling éloigné	ok	Ok	ok	ok	Lourd
Zoom	Non	Bon	?	?	La meilleure
Travelling+zoom	ok	Ok	?	?	Lourd

Tableau 2 : mini bilan des transformées

E. Proposition de mode de représentation et de visualisation

Le passage d'une vidéo (dynamique) vers une image (statique) provoque une perte importante de la notion de temporalité et de mouvement. Voici quelques propositions pour réintroduire la composante dynamique dans les images produites (certaines n'ont pas été implémentées).

1. Ajouter des symboles graphiques

- **Time-line**

L'ajout d'une frise, sorte de time-line peut aider à mesurer le défilement du cadre. A chaque nouvelle image, une barre blanche est marquée suivant les transformations. La plupart des images de ce rapport l'illustre.

Le tracking de points d'intérêt pourrait servir à tracer la trajectoire de quelques objets en mouvements locaux.

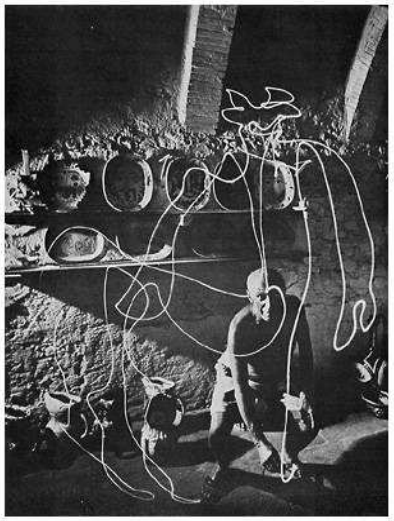


Image 22 : Mili Gjon, Picasso dans son atelier, procédé avec une lampe stroboscopique dans l'obscurité.

- **Notion de cadre :**

En plus de la perte de la notion temporelle, la plupart des mises à plat dissolvent l'intention de cadrage. C'est le cas du :

- modèle affine s'il y a une combinaison forte scale-rotation-échelle,
- zoom pur donnant l'impression de retrouver l'image la plus large de la séquence.

Je propose 2 simples ajouts de symboles graphiques : soit par une marque des coins des cadres en blancs, ou soit par une trace blanche du contour.

Image 23 : réintroduction de la notion de cadre

Avec ces traînées blanches rajoutées à l'exemple utilisé au B.2.b), nous pouvons maintenant comprendre que le cadre converge vers la porte.

2. Réinjecter une composante dynamique : « Motion mosaïc »

L'idée de réintroduire la vidéo m'est venue au cours d'un débuggage en visualisant une suite d'images isolée et recalée (je n'avais pas encore pris connaissance des travaux sur les « motion panoramas » de [BDH 03] (voir III.A.3.b)(3)), comme dans cet exemple d'une vue aérienne :

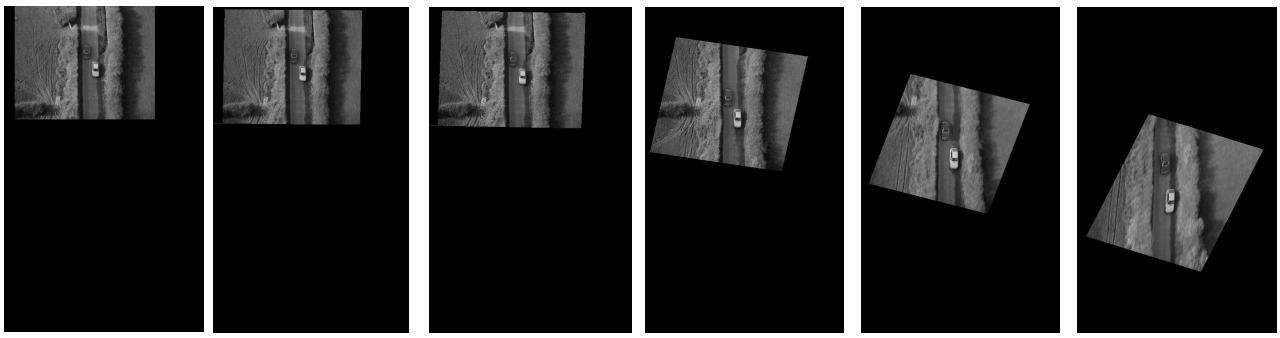


Image 24 :un ovni audiovisuel : la vidéo recalée

De la même manière qu'il est possible de voir la vidéo recalée dans le plan, nous pouvons imaginer deux autres modes de visualisation dérivés :

- réintroduction de la vidéo dans l'image mosaïque,
- réintroduction avec mise en évidence du cadre.

Image 25 : mosaïque animée d'un panoramique modélisé par une translation

Dans cet exemple, l'apport principal est dû à la réduction de la mosaïque par un modèle translation. Cette technique donne l'impression de passer une loupe rectangulaire et déformante sur l'image mosaïque, poussant les objets et les personnages à s'animer à son passage.
L'idée est de recréer une vidéo, pas une image. Nous pouvons imaginer une interface graphique de consultation de vidéos où les images mosaïques statiques s'animent au passage du curseur de la souris.

Si le modèle était perspectif, l'effet du procédé serait d'avoir une image presque statique, avec quelques petits objets s'animant de temps en temps. La notion de cadre est alors perdue.
Or, si nous nous inscrivons dans une problématique de consultation rapide de documents audiovisuels, nous devons faire ressortir le plus d'informations possibles sur l'image. Dans cet exemple, j'ai assombri tout simplement l'image mosaïque hors-cadre pour souligner l'intention du réalisateur.

Image 26 : séquence recalée avec mise en évidence du mouvement du cadre

Remarque : ces images peuvent être « consommées » telles quel. En balayant d'un coup d'œil ces 3 vignettes, nous percevons immédiatement l'action de la scène.

3. Mettre en évidence des mouvements par une segmentation au moindre coup

Si un recalage d'image est bien estimé, il est alors possible de profiter de cet outil pour effectuer une segmentation simplifiée.

Image 27 : Mili Gjon, surimpression

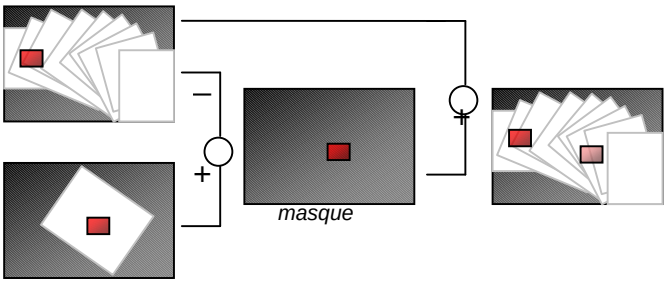


Figure 33 :segmentation et ajout d'objet fantôme

Image 28 : mosaïque avec objets fantômes

4. Ajout par tranches de taille variable

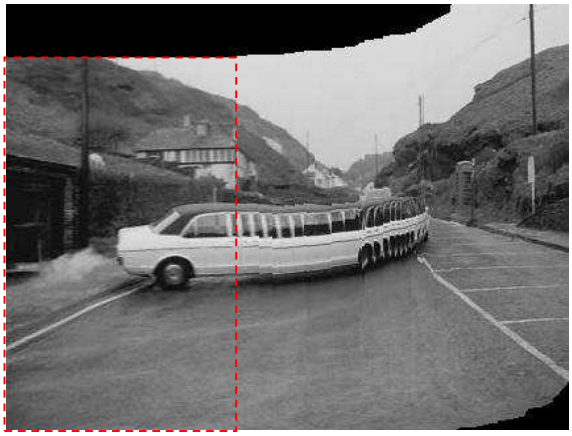


Image 29 : ajout par tranches à taille variable

Image 1

Le principe consiste à ajouter la première image en partie seulement, et d'effectuer l'assemblage par tranche de taille variables. Dans cet exemple, la fonction implémentée prend en paramètre la position en horizontal de coupe de la première image.

5. Ajout par fondu et fusion

L'ajout par fondu est souvent utilisé pour le « stiching » de photographies panoramiques pour adoucir les transitions entre les clichés. Dans notre cas, il semble inutile de fondre les transitions, souvent trop fines entre deux images successives d'une vidéo.
La fusion semble être une idée séduisante pour recaler les images, mais elle ne peut s'appliquer que dans le cas d'une mise à plat spatio-temporelle parfaite. Si le procédé fonctionne, les objets en mouvements locaux apparaissent comme des objets fantôme et rappelle le cas de la segmentation précédente. A l'inverse, un effet de flou apparaîtra.

F. Perspectives

1. A court terme

La priorité doit être donnée à l'implémentation de l'algorithme de Levenberg-Marquart pour raffiner les modèles non-linéaires. La bibliothèque de GSL contient toute une panoplie de fonctions pour le faire.

La seconde phase du projet devra s'orienter naturellement vers l'établissement d'une banque de tests pour valider, comparer et analyser finement les différents modèles en fonction de la nature de la scène audiovisuelle. Fort de ces expériences, une sélection automatique de modèle par apprentissage pourra être mise en oeuvre.

2. Intégration dans l'interface développée par l'équipe VIE

L'équipe VIE travaille entre autre sur la mise en place d'interfaces graphiques pour exploiter des arbres de données. L'utilisateur navigue dans une représentation circulaire et « zoomable », comme s'il était à l'intérieur des « branches », en partant du plus générique, le « tronc », et en arrivant aux « feuilles », c'est-à-dire les documents.

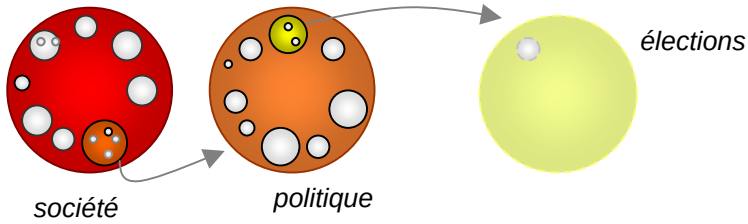


Figure 34 : interface de fouille d'arbre

Appliqué à une base de données de documents audiovisuels, un tel arbre mènerait naturellement en ses feuilles à la consultation de vidéo. L'interface pourrait alors prendre cette forme (voir Image 30).

- Analyse du document :

Le sens de lecture est implicite et culturel : il faut suivre les sens des aiguilles d'une montre de cette « horloge » pour suivre l'action du document. Avec cette représentation, nous observons rapidement que ces 9 premiers plans comportent :

- 2 longs travelling,
- un travelling optique (image 4),
- une suite de plans fixes en extérieur,
- et une alternance de plans larges et de gros plans en intérieur,
- un dernier plan fixe de dialogue.

Nous pouvons reconnaître une structure introductive d'une narration type documentaire avec une phase de mise en place du contexte en quelques plans avant de passer au cœur du sujet.

Il est ainsi possible de mettre en évidence le « comportement » narratif d'une séquence audiovisuelle. Voici un autre exemple où l'on perçoit assez rapidement, grâce aux formes variées des plans qu'il s'agit une scène d'action (une course poursuite en l'occurrence).

Image 30 : interface envisagée pour la consultation de vidéos

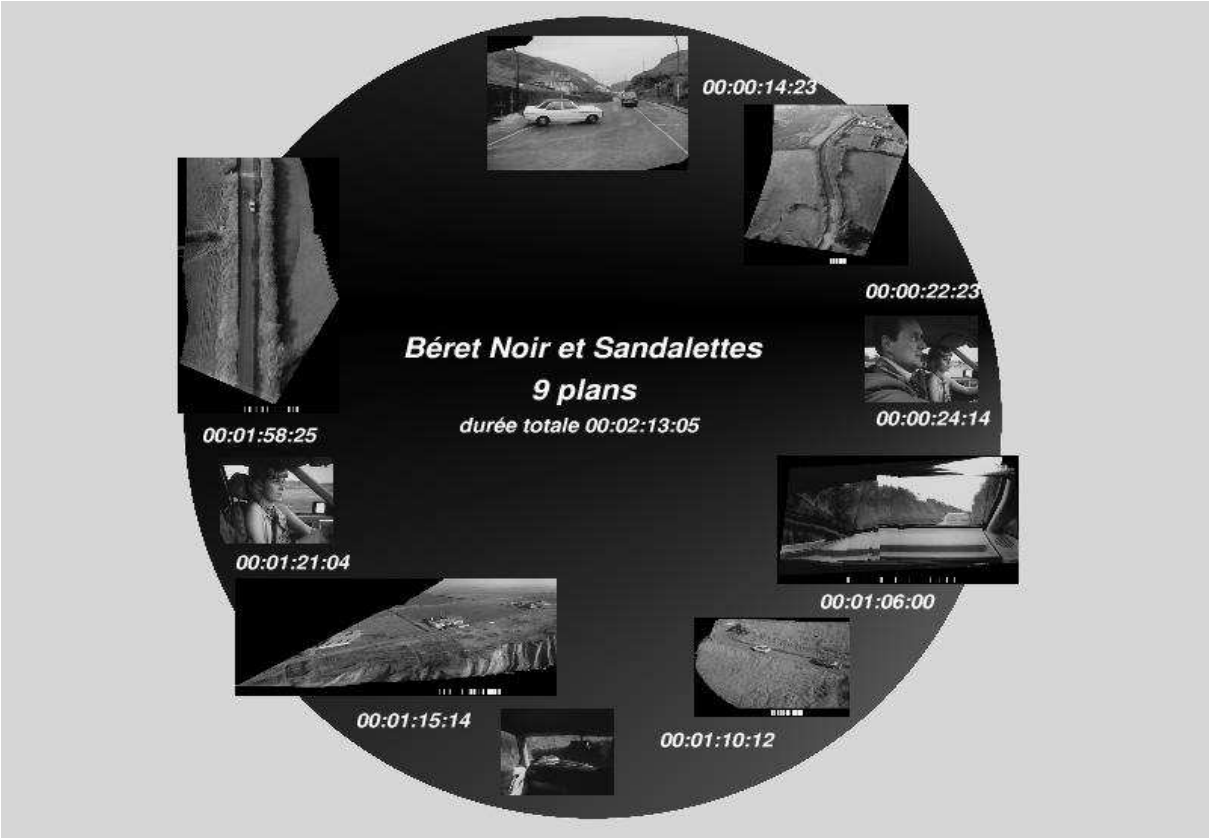


Image 31: visualisation d'une scène d'action

3. A long terme : réintroduire la notion temporelle par le son

A plus long terme, nous pourrions envisager d'employer le son. Quelques laboratoires travaillant sur l'interprétation des mouvements, dans des domaines aussi variés que la biomécanique ou la sismologie, traduisent les phénomènes physiques en un son audible. En effet, l'ouie est reconnue comme étant un sens plus fin que la vue pour percevoir de petites variations.

- **La métaphore du timbre :**

Une scène est composée d'un mouvement global, celui de la caméra, et de mouvements locaux, ceux des objets en mouvements. Tous ces déplacements peuvent être perçus comme un spectre fréquentiel : le mouvement de la caméra pourrait être traduit comme l'harmonique principale et les mouvements locaux comme les harmoniques du timbre.

Il s'agit de savoir dans quelle mesure il serait possible d'envisager un tel système de représentation. Après un certain apprentissage d'un « langage sonore », l'utilisateur pourrait intuitivement comprendre l'action d'une scène par son timbre.

BILAN

Au cours de ce stage, j'ai démontré qu'il était possible de réemployer la technologie de signature de l'INA pour obtenir des représentations originales de scènes audiovisuelles.

Les applications envisagées par la communauté scientifique en vision abordent souvent des problématiques plus attachées à la robotique qu'à l'audiovisuel. C'est donc un domaine qui mérite plus de réflexion, car il implique des cas plus riches et variés de prise de vue.

Dans ce rapport, nous nous sommes attardés sur la reconstruction d'images mosaïques ou panoramiques. Nous avons vu que plusieurs modèles étaient envisageables, de la simple translation au polynomial, avec leur lot d'avantages et d'inconvénients selon les différents types de prises de vues. La transformation affine offre le meilleur compromis entre rapidité et qualité de la mise à plat, les modèles polynomiaux sont plutôt lourds à gérer et la transformée de projection linéaire reste difficile à évaluer.

J'ai posé plusieurs problèmes liés à ce type d'application :

- l'interprétation de la profondeur de champ,
- la difficulté de réintroduire la notion temporelle,
- la dualité entre réduction temporelle et accroissement spatial de la vidéo,
- la difficulté d'implémenter un modèle unique pour toutes les prises de vues,
- l'interprétation des images : voulons nous voir une mise à plat parfaite ou garder la notion du cadre ?

Concernant la poursuite d'un tel travail, au-delà du perfectionnement technique du prototype, le projet nécessitera sûrement une analyse plus poussée de l'écriture de documents audiovisuels. Cela pourrait faciliter, la mise en place, par exemple, d'une aide à la décision de choix du modèle de visualisation. En effet, certains produits audiovisuels sont souvent stéréotypés comme le reportage d'information ou sportif, et dans une certaine mesure la fiction (conventionnelle comme le téléfilm…), voir certain documentaire.

C'est véritablement ma première expérience de programmation quotidienne en milieu professionnel. J'apprécie la créativité et la rigueur qu'implique le développement, et j'espère par la suite pouvoir me perfectionner dans ce domaine.

Plus généralement, mon projet professionnel est de pouvoir continuer à travailler dans un cadre de recherche et développement autour des images et des sons en s'appuyant sur une culture de l'Image.

VI. Annexes

1. Appareillage de positions

2. Estimation de transformés d'images

Tableau knn

Sélection aléatoire

Condition limites :
- poids minimum
- répartition des positions de l'image
- ensemble de positions différentes à chaque essai

Résolution de système linéaires (modèle affine)

Ajustement des moindres carrés

Distance euclidienne entre In et In' = T(In+1)

Conditions limites : θ_{max}, tx_{max}, ty_{max}, Sx_{max}, Sy_{max}

Distance entre points d'intérêt

Sélection de la transformée minimisant la distance

Sélection de la transformée possédant le plus d' « inliers »

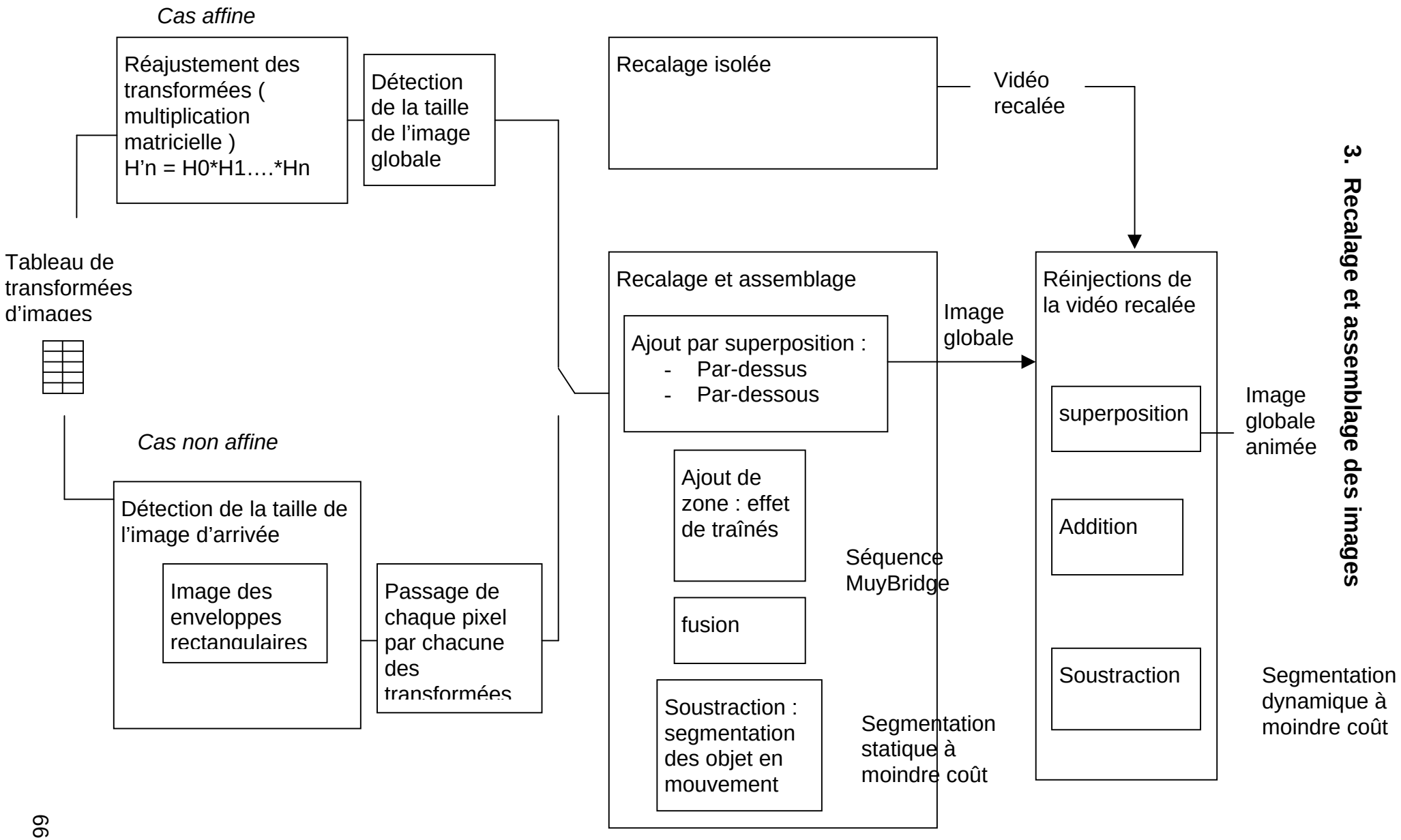
$T_{optimale}$

Tableau de transformées

In

In+1

In' = T(In+1)


Cas affine
Réajustement des transformées (multiplication matricielle)
H'n = H0*H1….*Hn
Détection de la taille de l'image globale
Tableau de transformées d'images
Cas non affine
Détection de la taille de l'image d'arrivée
Image des enveloppes rectangulaires
Passage de chaque pixel par chacune des transformées
Recalage isolée
Vidéo recalée
Recalage et assemblage
Ajout par superposition :
- Par-dessus
- Par-dessous
Ajout de zone : effet de traînés
fusion
Soustraction : segmentation des objet en mouvement
Séquence MuyBridge
Segmentation statique à moindre coût
Image globale
Réinjections de la vidéo recalée
superposition
Addition
Soustraction
Image globale animée
Segmentation dynamique à moindre coût

B. Information pratique sur l'environnement développé sur le poste Babylon

1. Utilisation du prototype

Les fichiers images doivent être au format .pgm avec numérotés avec un gabarit de 4 caractères (ex : sequence0000.pgm, sequence0001.pgm, …).

Commande :

```
./prototype prefixe_des_images numero_première_image numero_dernière_image pas_de_matching mode_de_selection_position_image_1 mode_de_selection_position_image_2 mode_matching taille_zone_recherche_matching mode_estimation_transformée nombre_coefficent_transformée nombre_essai mode_de_reconstruction nom_image_sortie
```

Ex: `./prototype seq 0 304 2 c g p 80 p 3 3000 g 1 seq_mis_a_plat`

Cette exemple permet la mise à plat de la suite d'image seq%04.pgm avec un pas de 2, c'est-à-dire que seq0000.pgm est matché avec seq0002.pgm, seq0002.pgm avec seq0004.pgm, …, jusqu'à l'image seq0304.pgm.

Modes disponibles :

argument 5 et 6:

- c : sélection des positions par le détecteur de Harris
- g : sélection des positions sur une grille (pas de 1 par défaut, mais réglable dans le fichiers GridSelection.C où l'on peut aussi activer une interface d'utilisation).

Argument 7 :

- p : matching sur des points
- b : bloc-matching (non réalisé)

Argument 9 :

- r : estimation de transformées d'image par l'algorithme Ransac
- p : estimation de transformées d'image par un algorithme « pseudo ransac »

Argument 12 :

- g : construction d'une image globale ajoutant par tranche les images de la séquence vidéo
- m : ajout sur l'image globale de la vidéo recalée
- s : soustrait pour arriver à une segmentation rudimentaire
- a : fusion des images

2. Utilisation de ffmpeg pour extraire des images pgm à partir d'une vidéo :

Ffmpeg –i nom_video.mpg –o nom_image%04.pgm

C. Glossaire

Ajustement multi-paramètres ou ajustement des moindres carrées : modélisation par une courbe d'un jeu de données éparses (voir aussi régression linéaire).

Appariement ou "Matching": association de données jugées proche. Dans notre cas nous faisons du « matching » de points d'intérêt, mais il est courant de l'effectuer avec des régions (block « matching » en MPEG2 et MPEG4).

Coin ou point d'intérêt : pixel à haute teneur en information. Point autour duquel son voisinage varie fortement en luminance dans plusieurs directions.

Descripteur : ensemble de valeurs caractérisant une entité. Dans notre cas des points d'intérêts de l'image sont associés à des vecteurs contenant des informations sur leur voisinage (quantité d'information, position,...)

Distance : toute donnée peut être associée à un vecteur. La distance entre deux données peut s'effectuer à partir de leurs vecteurs, et peut être ainsi significatif de leur ressemblance.

Estimation : en statistique, évaluation d'un modèle, d'une propriété mathématique liant un ensemble de points. L'estimation est d'autant plus robuste qu'elle tient en compte les points erronés (« outlier ») en privilégiant les « inliers ».

Feature: caractéristique d'un objet. En imagerie, cela peut être l'ensemble des points d'intérêt, les contours, les informations de couleurs…

Flot optique ou « optical flow »: détection de la variation d'intensité pour repérer la migration des pixels mouvement. Représentation en champs de vecteurs de dépalcement.

Homographie : lorsque deux images prises du même point de vue présentent une zone de recouvrement, elles sont alors liées par une transformée géométrique appelée homographie de la forme :

$$x' = \frac{ax + by + c}{mx + ny + p} \qquad y' = \frac{dx + ey + f}{mx + ny + p}$$

Images mosaïques : présentation de multiples images extraites d'une séquence vidéo, permettant un premier aperçu rapide du contenu.

Indexation : procédé associant un texte, une description, à une entité, dans notre cas des vidéos, permettant par la suite d'organiser l'ensemble des documents selon des thèmes ou corpus ou tout autre arborescence**.**

Invariance (à une transformation) : robustesse parfaite : maintient de tous les points d'intérêt après une transformation.

Knn (k nearest neighbour): lors d'association, de "matching" de points d'intérêt entre deux images, il est préférable de retenir plusieurs candidats, les « K plus proches voisins ».

Monitoring: système d'observation, « d'espionnage », de flux vidéos.

Morphing: généralisation de l'homographie à tout type d'image : il existe pour tout couple d'images une transformée les liants. Le morphing permet de décomposer en suite d'images interpolées le passage d'une image à une autre.

Occlusion : pertes de données dans l'image due à un chevauchement d'un objet sur un autre (ex: voiture passant devant une personne).

Outlier/inlier: données considérées conformes ou non à une propriété déduite par la modélisation de l'ensemble des données.

Recalage d'images : mise à plat d'images issues d'une même scène.

Régression linéaire : modélisation par une droite d'un jeu de données plus ou moins éparses. Le résultat est souvent obtenu par la minimisation de la somme pondérée des distances résiduelles au carrée.

Robustesse (dans le contexte d'une estimation de mouvement) : résistance à l'estimation de données aberrantes. Autrement dit, c'est la capacité à garder seulement les points d'intérêt pertinents.

Segmentation d'image : isolement d'une ou plusieurs régions caractérisées par une homogénéité (par exemple l'intensité, la texture, etc). Il existe deux grandes familles de segmentation par région ou par contour.

Signature (d'images) : technique se basant sur des descripteurs de points d'intérêt pour caractériser et décrire une image de manière unique. Ainsi une technique de signature performante permet de retrouver une séquence vidéo à partir de la comparaison des signatures d'images qu'elle contient.

Skimming (ou écumage) : thème de recherche en visualisation vidéo, dont l'objectif est d'épurer le contenu au maximum, afin de présenter seulement les moments clés.

Stiching: procédé popularisé par l'avènement de la photographie numérique pour reconstituer un panoramique à partir de plusieurs clichés pris à partir du même point de vue.

Vote : attribution d'un poids à chaque données issue d'un ensemble selon un critère. Dans notre cas, une première attribution de poids est effectuée pour chaque knn selon un critère de distance entre chaque position.

D. Références

[BDH 03] Andrien Bartoli, Navneet Dalaal, Radu Horaud, *Motion Panoramas*, Rapport de recherche INRIA, mars 2003.

[CZ] David Capel, Andrew Zisserman, *Automated Mosaicing with Super-resolution Zoom*

[FB 81] M.A. Fischer, R.C Bolles, *Random Sample Consensus: A paradigm for Model Fitting with Applications To Image Analysis and Automated Cartography,* Comm. Of the ACM, vol 24, pp381-395, 1981.

[JFB 03] Alexis Joly, Carl Frelicot, Olivier Buisson, *Robust content-based video copy identification in a large reference database,* CIVR *2003.*

[Lal 01] France Laliberté, *Recalage et fusion de fond d'œil*, Centre de Recherche et d'Informatique de Montréal, mai 2001.

[MC 01] K. Mikolajczyk and C. Schmid, *Indexation à l'aide de points d'intérêt invariants à l´echelle,* In Orasis, 77-86, juin 2001.

[PM] Satya Prakash Mallick, *Feature Based Image Mosaicing*

[SK99] Christoph Stiller, Janusz Konrad, *Estimating Motion in Image Sequences, a tutorial on modeling and computation of 2D motion, IEEE Signal Processing Magazine,* July 99.

[SSO] Aljoscha Smolic, Thomas Sikora, Jens-Rainer Ohm, *Direct Estimation Of Long-Term Global Motion Parameters Using Affine And Higher Order Polynomial Models*

[Ste 99] Charles V.Stewart, *Robust Parameter Estimation in Computer Vision, SIAM Review*, Vol. 41, No. 3, pp. 513-537, 1999.

[TZ 99] P.H.S Torr and A. Zisserman, *Feature Based Methods for Structure and Motion Estimation*,

[Zhe03] Jiang Yu Zheng, *Digital Route Panoramas, IEEE Multimédia 2003*

[HZ 00] R.Hartley and A.Zisserman, *Multiple View Geometry in Computer Vision,* Cambridge University Press, Cambridge, Uk, 2000

site :
[Sil] http://www.enseignement.polytechnique.fr/profs/informatique/Francois.Sillion/Majeure/Projets/teyssier/projet.html

E. Table des illustrations

1. Table des figures

2. Tableaux

3. Table des images

F. Index