

# Diallo Alpha Oumar Binta – Rectification Plane d'Image (TER)

---



*Nom – Prénom : Diallo Alpha Oumar Binta*

*Année d'étude : 2011-2012*

*Période : 16 Avril 2012 – 8 Juin 2012*

*Encadrant du TER : Alain Crouzil*

*Sujet de TER : Rectification plane d'image*

*Licence 3 Informatique parcours Ingénierie des Systèmes Informatiques*

*Université Paul Sabatier Toulouse*



# Diallo Alpha Oumar Binta – Rectification Plane d’Image (TER)

---

## TABLE DES MATIÈRES

1	REMERCIEMENTS.....	5
2	INTRODUCTION .....	6
3	PRESENTATION DU CONTEXTE DU TER .....	7
3.1	Le laboratoire .....	7
3.2	L’équipe TCI (Traitement et compréhension d’image) .....	8
3.3	Cadre de l’étude .....	8
3.4	Moyens mis à disposition .....	8
4	PRESENTATION DU SUJET.....	9
4.1	Problème informatique à résoudre.....	9
4.2	Objectifs.....	9
4.2.1	Besoins fonctionnels.....	9
4.2.2	Besoins non fonctionnels .....	10
4.3	Outils informatiques utilisées.....	10
4.3.1	Solutions envisageables.....	10
4.3.2	Solution retenue.....	11
4.3.3	Présentation succincte de java .....	11
4.3.4	Présentation succincte de jama .....	11
5	DEMARCHE METHODOLOGIQUE.....	12
6	CONCEPTION ET REALISATION DE L’APPLICATION.....	13
6.1	Analyse du problème.....	13
6.2	Différentes solutions techniques possibles.....	13
6.3	Choix retenu .....	13
6.4	Rectification de l’image .....	14
6.4.1	Principe.....	14
6.4.2	Etapes de la rectification plane d’une image .....	14
6.4.3	Notion d’homographie .....	14
6.4.4	Application de la transformation .....	17
6.4.5	Transport inverse des pixels.....	17
6.5	L’interface graphique .....	21
6.5.1	Éléments graphiques utilisées.....	21

# Diallo Alpha Oumar Binta – Rectification Plane d’Image (TER)

---

6.5.2	Explications des zones de l’interface graphique .....	22
6.5.3	Exemple de Boîtes de dialogues utilisées.....	24
7	CONCLUSION .....	25
8	BIBLIOGRAPHIE.....	26

Figure 1	Rectification plane d'une image .....	9
Figure 2	Exemple de rectification selon les choix de l'utilisateur .....	16
Figure 3	Transformation géométrique d'une image discrète .....	17
Figure 4	Algorithme de construction de l'image transformée .....	18
Figure 5	Interpolation bilinéaire .....	19
Figure 6	Domaine de modélisation de l'interpolateur bicubique .....	19
Figure 7	Exemple de rectification plane d'une image selon les deux choix possibles .....	20
Figure 8	Exemple de rectification selon la taille de l'objet .....	20
Figure 9	Proposition d'interface graphique .....	21
Figure 10	Interface graphique de l'application .....	22
Figure 11	Paramétrage de la rectification d'une image .....	23
Figure 12	Boîte de dialogue pour ouvrir un fichier .....	24
Figure 13	Boîte de dialogue pour entrer les informations nécessaires pour effectuer la transformation .....	24
Figure 14	Exemple de message d'erreur .....	24

Tableau 1	Tableau comparatif des solutions technologiques envisageables .....	10
Tableau 2	Planning prévisionnel pour le projet .....	12

Équation 1	$H$ homographie de $\mathbb{R}^2$ dans $\mathbb{R}^2$ .....	14
Équation 2	Correspondance de deux points .....	15
Équation 3	Ecriture matricielle pour la détermination des paramètres de l'homographie .....	15
Équation 4	Passage de nouvelles coordonnées aux anciennes .....	17
Équation 5	Interpolation bilinéaire d'un point .....	18
Équation 6	Interpolation bicubique d'un point .....	20

## 1 REMERCIEMENTS

Je remercie sincèrement Monsieur Alain Crouzil, maître de conférence à l’Université Paul Sabatier qui m’a accompagné tout au long de cette expérience avec beaucoup de patience, de pédagogie et de m’avoir trouvé ce sujet de TER. Sa motivation constante et ses conseils m’ont été très précieux.

Je tiens toute l’équipe pédagogique de l’Université Paul Sabatier et les intervenants professionnels responsables de la formation **ISI (ingénierie des systèmes informatiques)**, pour avoir assuré la partie théorique de celle-ci.

## 2 INTRODUCTION

Afin de valider ma troisième et dernière année de licence, j’ai effectué un **TER** d’une durée de sept semaines sous la direction de monsieur Alain Crouzil, maître de conférence à l’université Paul Sabatier de Toulouse, membre de l’équipe **TCI** au sein de l’**IRIT**.

Ce TER, se situe dans le cadre de la vision par ordinateur qui est un domaine de recherche de l’équipe TCI. La rectification plane d’une image consiste à produire une image simulant une prise de vue de face où le plan image est parallèle au plan de la scène. Cela peut être utile dans plusieurs applications dans lesquelles il n’est pas facile de se placer en face de l’objet. Un exemple serait la photographie de bâtiments, de tableaux ou la numérisation de document.

L’objectif de ce TER est de commencer le développement d’une application permettant de rectifier une image en utilisant différentes méthodes proposées par des chercheurs de l’équipe. Il s’agit également de concevoir et d’implémenter une interface graphique de cette application. A long terme, l’application sera destinée à être modifiée, complétée et devra permettre de réaliser une comparaison de performance des méthodes implémentées pour la rectification plane d’une image.

Je vous expose dans ce rapport en premier lieu une présentation du laboratoire, son organisation et ses activités. J’aborderai ensuite les éléments nécessaires à la compréhension de ce TER, la mise en œuvre concrète de ce travail étape par étape de l’analyse à la conception en évoquant les problèmes rencontrés ainsi que les moyens d’y remédier, puis je présenterais en conclusion le bilan des résultats obtenus, les perspectives d’évolution, les apports de ce stage par rapport aux enseignements reçus dans le cadre de ma formation et à mon projet professionnel.

## 3 PRESENTATION DU CONTEXTE DU TER

### 3.1 LE LABORATOIRE

L'IRIT (Institut de Recherche en Informatique de Toulouse) est une unité mixte de recherche UMR 5505 (Unité Mixte de Recherche). Il a été fondé en 1990 en partenariat entre l'Université Paul Sabatier de Toulouse (UT3), le CNRS (Centre National de la Recherche Scientifique), l'ENSEEIH (Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications), l'Institut national polytechnique de Toulouse et l'Université des Sciences Sociales de Toulouse (UT1) devenue en 2009 Université Toulouse 1 Capitole. Il est issu de la fusion de deux URA (Unités de Recherche Associées) du CNRS et de l'Université Paul Sabatier (LSI : Langages et Systèmes Informatiques, dirigé par René Beaufils puis Jean Vignolle et le CERFIA (Centre d'Etudes sur la Reconnaissance des Formes et l'Intelligence Artificielle), dirigé par Serge Castan et Guy Pérennou) et du CIT (Centre Informatique de Toulouse) de l'Université Paul Sabatier, créé par Michel Laudet. Il a été dirigé de 1998 à 2011 par Luis Farinas Del Cerro, et est dirigé depuis juillet 2011 par Michel Daydé. L'IRIT représente un des plus forts potentiels de la recherche en informatique en France avec un effectif global de 600 personnes dont 250 chercheurs et enseignants-chercheurs, 244 doctorants, 14 Post-Doctorants et chercheurs contractuels ainsi que 43 ingénieurs et administratifs. Les recherches sont organisées autour de sept thèmes :

1. Analyse et synthèse de l'information (Responsable: Philippe Joly Equipes: SAMoVA, SC, TCI et VORTEX)
2. Indexation et recherche d'informations (Responsable: Franck Ravat Equipes: PYRAMIDE et SIG)
3. Interaction, autonomie, dialogue et coopération (Responsable: Pascale Zaraté Equipes IC3, ELIPSE et SMAC)
4. Raisonnement et décision (Responsable: Andreas Herzig Equipes: LILaC et ADRIA)
5. Modélisation, algorithmes et calcul haute performance (Responsable: Jean-Marc Alliot Equipe: APO)
6. Architecture, systèmes et réseaux (Responsable: Christine Rochange Equipes IRT, SEPIA, SIERA, T2RS et TRACES)
7. Sécurité de développement du logiciel (Responsable: Christian Percebois Equipes ACADIE, ICS et MACAO)

## 3.2 L’ÉQUIPE TCI (TRAITEMENT ET COMPRÉHENSION D’IMAGE)

Le domaine de recherche de l’équipe TCI est l’analyse d’image, c’est-à-dire le traitement et l’interprétation de données provenant d’un ou de plusieurs systèmes d’acquisition d’image. Leurs travaux couvrent trois domaines :

1. la vision par ordinateur.
2. l’imagerie médicale
3. la langue des signes.

## 3.3 CADRE DE L’ÉTUDE

Ce TER se situe dans le cadre de la mise en correspondance de pixels pour la stéréovision binoculaire, un thème de recherche sur lequel travaille monsieur Alain Crouzil au sein de l’équipe TCI. La mise en correspondance de pixels consiste à trouver dans deux images d’une même scène, prises à des positions différentes, les couples de pixels qui sont les projections d’un même point de la scène. L’objectif du TER est de commencer l’implémentation des algorithmes permettant l’estimation des paramètres permettant d’effectuer une transformation géométrique sur une scène comportant un plan de telle sorte, que dans l’image résultat le plan semble être photographié de face, c’est-à-dire parallèle au plan image puis de concevoir et implémenter une interface graphique de l’application. Cette opération présente un intérêt dans de nombreuses applications :

- La numérisation de documents (pour remplacer un scanner à plat par un appareil photographique ou un “Smartphone”).
- La photographie de bâtiments ou de tableaux.

## 3.4 MOYENS MIS À DISPOSITION

Pour la réalisation de ce TER, j’avais sous la main un fichier montrant les différentes étapes à suivre pour la rectification plane d’une image en utilisant une homographie.



## 4 PRESENTATION DU SUJET

### 4.1 PROBLÈME INFORMATIQUE À RÉSOUDRE

Un objet plan photographié à l'aide d'un appareil qui n'est pas parfaitement en face de l'objet, l'effet de perspective induit dans l'image obtenue des propriétés géométriques parfois indésirables rendant difficile son utilisation. Les droites parallèles sur l'objet sont concourantes sur l'image. Pour une bonne utilisation de cette image, une transformation géométrique est nécessaire. Cette opération géométrique a pour but de modifier la position des informations contenues dans l'image sans modifier le niveau de gris. Cette opération peut s'appliquer à la totalité des points de l'image (les pixels) ou à un objet particulier de l'image.

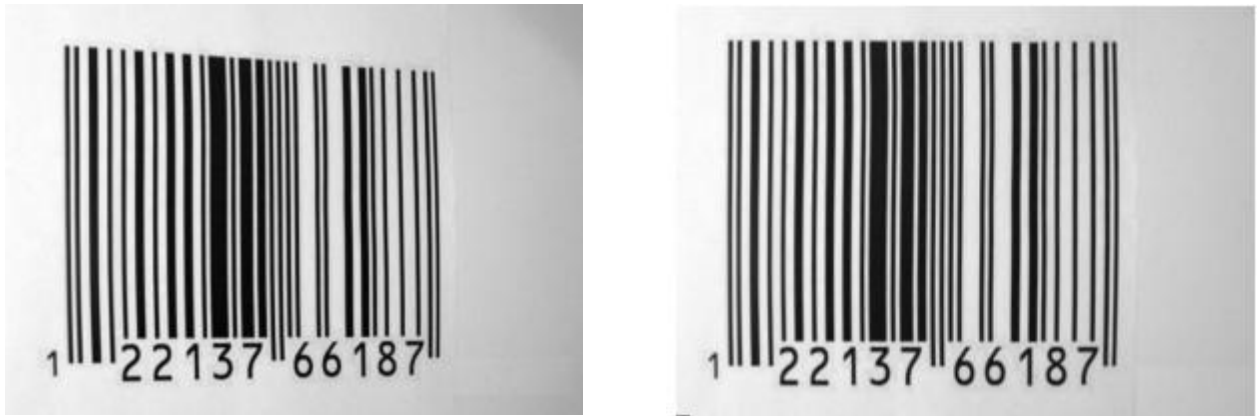


Figure 1 Rectification plane d'une image

### 4.2 OBJECTIFS

Le travail qui m'a été demandé par monsieur Alain Crouzil est d'implémenter des algorithmes permettant d'estimer les paramètres de la transformation géométrique d'une scène comportant un plan de telle sorte, que dans l'image résultat le plan semble être parallèle au plan image. Il s'agit également de concevoir et d'implémenter une interface graphique de l'application.

#### 4.2.1 BESOINS FONCTIONNELS

L'application doit être en mesure :

1. D'ouvrir plusieurs sortes de formats d'images.
2. De permettre à l'utilisateur de sélectionner des points et des droites dans une image.
3. De permettre à l'utilisateur d'effectuer un zoom une image.
4. De dessiner un quadrilatère afin de créer une zone de sélection.

# Diallo Alpha Oumar Binta – Rectification Plane d’Image (TER)

---

5. De permettre la rectification plane d’une image et d’afficher le résultat de la rectification.
6. De permettre à l’utilisateur un choix de transformation :
  - i) L’image rectifiée aura les mêmes dimensions que l’image initiale et que le document occupera au maximum la surface de l’image rectifiée.
  - ii) L’image rectifiée aura les mêmes dimensions que l’image initiale.

## 4.2.2 BESOINS NON FONCTIONNELS

L’application réalisée doit être :

- Rapide au niveau performance
- Facilement maintenable
- Convivial et évolutif
- Portable (c’est-à-dire fonctionnel sous tous les systèmes d’exploitation).

## 4.3 OUTILS INFORMATIQUES UTILISÉES

### 4.3.1 SOLUTIONS ENVISAGEABLES

Plusieurs solutions technologiques étaient possibles. Il s’agit de lister les solutions à la fois portables et simples à mettre en œuvre. Les solutions de langage de programmation sont alors les suivantes :

Solution	Application exécutable	Avantages	Inconvénients
« 100 % C »	C	<ul style="list-style-type: none"><li>• Rapidité</li><li>• Simplicité de codage</li></ul>	<ul style="list-style-type: none"><li>• Interface spécifique à chaque plate-forme</li><li>• Portage difficile entre systèmes</li></ul>
« 100 % C++ »	C++	<ul style="list-style-type: none"><li>• Rapidité</li></ul>	<ul style="list-style-type: none"><li>• Idem « 100% c »</li></ul>
« 100 % Java »	Java	<ul style="list-style-type: none"><li>• Affranchissement des plates-formes (IHM et portage)</li><li>• Simplicité de codage</li></ul>	<ul style="list-style-type: none"><li>• Dégradation de la vitesse d’exécution (Java semi compilé)</li></ul>

Tableau 1 Tableau comparatif des solutions technologiques envisageables

## 4.3.2 SOLUTION RETENUE

Le choix technologique s'est rapidement porté sur une solution homogène, portable avec une mise en œuvre aisée. Nous avons donc retenu la solution « 100 % Java » car elle offrait tous ces avantages. Pour le calcul matriciel, j'ai utilisé la bibliothèque JAMA. La gestion des images est assurée par des instances de la classe BufferedImage de JAVA.

## 4.3.3 PRÉSENTATION SUCCINCTE DE JAVA

Java est un langage de programmation développé par Sun au milieu des années 1990 avec comme exigence principale la portabilité poussée à l'extrême. Ce langage implémente entièrement les concepts de la programmation objet : encapsulation, polymorphisme, héritage. A la différence des langages comme le C ou le C++, Java ne fournit pas un code directement exécutable par la machine mais un « byte-code » (code intermédiaire) qui sera exécuté par une machine virtuelle. Cette dernière est une application portée sur la majorité des plates-formes existantes (Windows, UNIX, Solaris) qui interprète le byte-code. C'est grâce à cette propriété qu'un programme Java s'exécute de la même manière quelle que soit la plate-forme. Il met à notre disposition un ensemble de package graphique rendant simple la création d'interfaces utilisateurs en vue d'une interactivité avec le logiciel et une intuitivité des plus accrues. Un gestionnaire intégré à Java permet la création d'API permettant une reprise optimale du logiciel pour des modifications plus aisées. Il est beaucoup plus facile d'obtenir un code sans erreur avec Java qu'avec C++ car un certain nombre de points susceptibles d'apporter des erreurs tel que :

- L'allocation et la libération de mémoire manuelles ont été retirées. En effet la mémoire, dans Java, est allouée et libérée automatiquement.
- Java permet, entre autres, une vraie gestion des tableaux car les concepteurs ont retiré l'arithmétique des pointeurs. La notion de référence sur une zone mémoire remplace avantageusement celle de " pointeur ", car elle supprime la possibilité d'écraser toute zone mémoire à cause d'un compteur erroné.
- Les concepteurs ont supprimé l'héritage multiple en le remplaçant par une nouvelle notion d'interface dérivée d'Objective C. Une interface nous offre les mêmes possibilités que l'héritage multiple, sans la complexité de la gestion de hiérarchie d'héritage multiple.

## 4.3.4 PRÉSENTATION SUCCINCTE DE JAMA

[JAMA](#) est un package de base d'algèbre linéaire pour Java. Il fournit à l'utilisateur des classes pour construire et manipuler des réels et des matrices denses. Il est destiné à servir de la classe matrice standard pour Java.

## 5 DEMARCHE METHODOLOGIQUE

Tout projet se doit de démarrer par le choix de la méthode de travail qui conditionnera toute la démarche, pour cela il a fallu définir les étapes essentielles et estimer le temps à consacrer pour chacune. Pour cela, j’ai établi un planning afin de représenter les différentes tâches à réaliser.

Semaine	Tâche à réaliser
1	Analyse des besoins et faisabilité
2	Choix du langage de programmation et spécification
3	Conception architecturale
4	Conception détaillée
5	Codage
6	Test et correction
7	Création de la documentation de l’application et Rédaction du rapport

*Tableau 2 Planning prévisionnel pour le projet*

La communication est indispensable à la réussite d’un projet. Afin de situer l’avancement du projet mon tuteur et moi échangeons régulièrement des mails et nous discutons de certains aspects lors des réunions, un **wiki** (Un wiki est un site web dont les pages sont modifiables par les visiteurs afin de permettre l'écriture et l'illustration collaboratives des documents numériques qu'il contient) avait été spécialement dédié au projet pour effectuer un rapport journalier sur le travail réalisé.

## 6 CONCEPTION ET REALISATION DE L'APPLICATION

### 6.1 ANALYSE DU PROBLÈME

La structuration du problème est soumise d'une part à l'implémentation d'algorithme de rectification plane d'image et d'autre part à la conception de l'interface graphique de l'application.

Pour la réalisation de ce TER, trois phases ont été définies. La première consiste à réaliser la recherche des outils nécessaires pour la gestion d'image et le traitement matriciel.

La seconde phase consiste à implémenter les algorithmes de rectification plane d'une image et de créer le noyau du programme permettant les interactions entre l'interface graphique et les données. Le noyau se composera d'un ensemble de petits programmes simplifiant la réalisation de la troisième et dernière phase.

Cette dernière phase consiste à créer l'interface graphique. Dans un premier temps, une étude sera réalisée pour présenter une interface plus efficace en établissant une charte graphique. Cette phase se terminera par la programmation de l'interface graphique en elle-même.

### 6.2 DIFFÉRENTES SOLUTIONS TECHNIQUES POSSIBLES

Les paramètres de la rectification peuvent être déterminés de plusieurs manières différentes. Ils peuvent par exemple être calculés si l'on connaît :

1. Les dimensions d'un rectangle situé sur l'objet.
2. Deux points de fuite correspondants à deux familles orthogonales de droites parallèles.
3. Certaines valeurs que les appareils photographiques numériques modernes incluent dans les images.

### 6.3 CHOIX RETENU

Il n'y avait pas de choix définitif car l'objectif de ce TER est de commencer l'implémentation des algorithmes et de créer l'interface graphique de l'application. Le choix retenu pour ce TER est la première ci-dessus, rectifier l'image connaissant les dimensions d'un rectangle situé sur l'objet à partir d'une homographie. Pour la gestion de l'image, mon choix a porté sur la classe `BufferedImage` qui hérite de la classe `Image` (optimisation de l'utilisation de la mémoire et permet de faciliter la sauvegarde de l'image) et le traitement matriciel est assuré par la bibliothèque JAMA (voir page 11).

## 6.4 RECTIFICATION DE L'IMAGE

### 6.4.1 PRINCIPE

Lorsqu'un objet photographié est plan, les points de ce plan sont liés à leurs projections sur l'image par une transformation projective appelée **Homographie**. Il est alors possible de calculer l'image que l'on aurait obtenue en changeant l'orientation de la caméra en se plaçant face au plan, dans une configuration fronto-parallèle dans laquelle le plan de la scène est parallèle au plan de l'image. Pour rectifier l'image, il faut déterminer les coefficients de la transformation qui amène les points de l'image source plane à une position de référence.

### 6.4.2 ETAPES DE LA RECTIFICATION PLANE D'UNE IMAGE

La rectification plane de l'image se déroule en sept étapes :

1. Ouverture de l'image source en niveau de gris.
2. Création de la matrice de l'image rectifiée ayant la même taille que l'image source.
3. Localisation dans l'image source de la position de quatre points.
4. Fixation des positions de leurs correspondants dans l'image rectifiée.
5. Estimation de la matrice homographique.
6. Détermination de la matrice de transformation finale (nécessaire si on veut avoir une image de même taille que l'image source).
7. Application de la transformation géométrique.

### 6.4.3 NOTION D'HOMOGRAPHIE

**Définition :** Une homographie  $H$  est une application de  $\mathbb{R}^2$  dans  $\mathbb{R}^2$ , définie par :

$$H : (x, y) \leftrightarrow (x', y') = \begin{cases} x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{cases}$$

$$\alpha \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \leftrightarrow \alpha P' = HP \quad \forall \alpha \neq 0$$

Équation 1  $H$  homographie de  $\mathbb{R}^2$  dans  $\mathbb{R}^2$

#### Propriétés

- Si  $\det(H) \neq 0$  alors  $\mathcal{H}$  est une bijection de  $\mathbb{R}^2 - \{(x, y) \mid h_{31}x + h_{32}y + h_{33} = 0\}$  Vers

# Diallo Alpha Oumar Binta – Rectification Plane d'Image (TER)

---

$$\mathbb{R}^2 - \{(x', y') \mid (h_{21}h_{32} - h_{22}h_{31})x' + (h_{12}h_{31} - h_{11}h_{32})y' + (h_{11}h_{22} - h_{12}h_{21}) = 0\}$$

- $H$  est défini à un facteur près et seulement 8 paramètres sont indépendants (sur 9 éléments).

## 6.4.3.1 ESTIMATION DES PARAMÈTRES DE L'HOMOGRAPHIE

Le problème revient à déterminer la matrice homographique à partir de correspondance de point 2D. Ces points sont liées entre eux par l'Homographie de la forme :  $P' = HP$  avec  $P$  et  $P'$  points en correspondance de coordonnées  $(x, y, z)^T$  et  $(x', y', z')^T$  (la dimension  $z'$  est fixée à 1 pour l'image plane). L'Homographie  $H$  est caractérisée par une matrice homogène 3x3. Cette transformation induit un facteur d'échelle et ne comporte que 8 coefficients indépendants ( $h_{33} = 1$ ). Une correspondance donne 2 équations :

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

$$\begin{cases} x'(h_{31}x + h_{32}y + h_{33}) = (h_{11}x + h_{12}y + h_{13}) \\ y'(h_{31}x + h_{32}y + h_{33}) = (h_{21}x + h_{22}y + h_{23}) \end{cases}$$

*Équation 2 Correspondance de deux points*

Avec  $n$  correspondance, on obtient un système homogène (second membre nul) de  $2n$  équations :

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y'_i & -y_i y'_i & -y'_i \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$A_{2n \times 9} = \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y'_i & -y_i y'_i & -y'_i \end{pmatrix}$$

*Équation 3 Ecriture matricielle pour la détermination des paramètres de l'homographie*

**Estimation** – On utilise  $n \geq 4$  correspondances de points dont les positions sont bruitées. Le système est alors surdéterminé : il n'admet pas de solution. On procède à une estimation au sens des moindres carrés totaux, c'est-à-dire que l'on calcule :

$$\hat{H} = \operatorname{argmin} \|AH\|^2$$

$\hat{H}$  = vecteur propre unitaire associé à la plus petite valeur propre de  $A^T A$ .

# Diallo Alpha Oumar Binta – Rectification Plane d’Image (TER)

---

## 6.4.3.2 CORRESPONDANCE DE POINT

### Localisation dans l’image initiale d’au moins 4 points

A ce niveau j’ai mis en place un outil de sélection précise de point pour permettre à l’utilisateur de sélectionner les quatre coins d’un objet situé sur l’image.

### *Correspondants dans l’image rectifiée des points localisés par l’utilisateur :*

Dans le cas de 4 points qui forment un rectangle dans la scène, il suffit de connaître le rapport entre les deux dimensions du rectangle dans la scène et de décider des positions dans l’image rectifiée de deux extrémités de l’un des côtés du rectangle projeté. Des dimensions classiques sont proposées à l’utilisateur, il peut aussi fournir les dimensions de l’objet ou le rapport entre les deux dimensions de l’objet.

**Choix de la rectification à appliquer :** Deux choix de rectification sont proposés :

1. l’image rectifiée aura la même taille que l’image initiale (image au centre)
2. l’objet occupera au maximum la surface de l’image rectifié (image à droite)



Figure 2 Exemple de rectification selon les choix de l’utilisateur



## 6.4.4 APPLICATION DE LA TRANSFORMATION

### 6.4.4.1 TRANSPORT DIRECT DES PIXELS

La mise en œuvre des opérations géométriques sur les images demande une algorithmique spécifique. En effet, l'idée de "construire" la nouvelle image à partir du balayage de l'original ne permet pas d'obtenir une image correcte. En effet, la transformation des coordonnées d'origines discrètes (donc entières) peut donner des valeurs non-entieres dans le résultat du calcul  $(X_{A'}, Y_{A'}) = H(X_A, Y_A)$ . La position d'un tel point est donc hors de la maille d'échantillonnage (fig. 3).

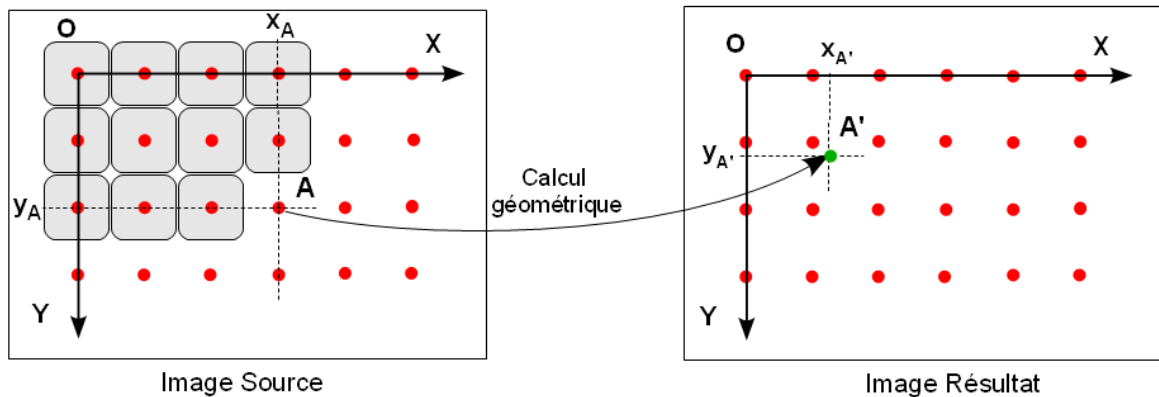


Figure 3 Transformation géométrique d'une image discrète

La solution d'affecter le résultat au pixel le plus proche par exemple ne résout pas le problème de la construction de l'image résultat; certains pixels seraient affectés de deux valeurs différentes, d'autres d'aucune valeur (trous et conflits).

### 6.4.5 TRANSPORT INVERSE DES PIXELS

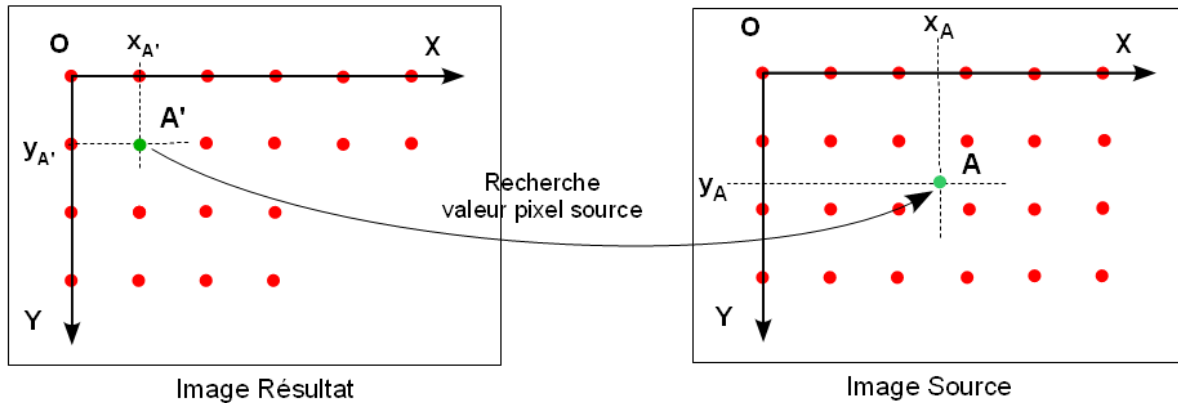
La méthode est de balayer les points de l'image résultat, dont les coordonnées sont dans la maille d'échantillonnage du résultat et à rechercher le niveau de gris équivalent du point source (fig. 4). A noter que la maille d'échantillonnage du résultat pourra être différente de celle de l'image d'origine dans le cas d'un *ré-échantillonnage*. En outre, l'antécédent d'un pixel peut ne pas être sur l'image initiale. Dans ce cas, on choisit une valeur de niveau de gris (généralement 0).

L'opération géométrique permettant de repasser des nouvelles coordonnées aux anciens est l'inverse de l'opération directe de la forme  $(X_A, Y_A) = H^{-1}(X_{A'}, Y_{A'})$  soit :

$$\begin{cases} x_A = H^{-1}x_{A'} \\ y_A = H^{-1}y_{A'} \end{cases}$$

Équation 4 Passage de nouvelles coordonnées aux anciennes

Plusieurs méthodes sont proposées pour interpoler la valeur du niveau de gris de A.



*Figure 4 Algorithme de construction de l'image transformée*

### 6.4.5.1 INTERPOLATION DU PLUS PROCHE VOISIN

C'est l'algorithme le plus simple, rapide à calculer. La fonction mise en œuvre est celle de l'arrondi (round, floor ou ceil). L'algorithme utilisé est :

```

For x = 1 à Xmax
  For y = 1 à Ymax
    x1 = round (HX)
    y1 = round (HY)
    If (x1 >= 1 & y1 >= 1 & x1 < Xmax & y1 < Ymax)
      Image_resultat(x, y) = Image_source(x1, y1)
    End
  End
End
    
```

*La qualité du résultat est assez faible : le phénomène de crénelage est très visible en cas de rotation d'un élément à aspect géométrique.*

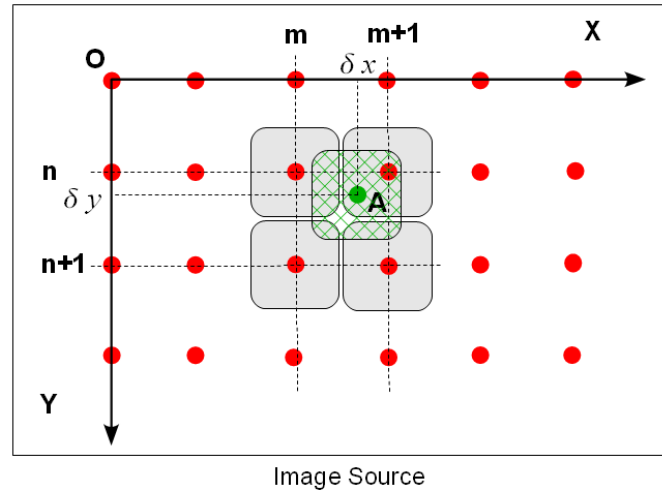
### 6.4.5.2 INTERPOLATION BILINÉAIRE

L'interpolation bilinéaire prend en compte pour le calcul les 4 pixels les plus proches du pixel inconnu, en donnant un poids évalué à partir de la distance du point calculé aux voisins. Soient (m, n) les coordonnées du point P<sub>0</sub> le plus proche obtenues par arrondi par valeur inférieure (floor) et (δx, δy) les coordonnées relatives du point A par rapport au point P<sub>0</sub>. L'interpolation du niveau de gris par la loi bilinéaire est donnée par :

$$I_A = (1 - \delta x)(1 - \delta y)I(m, n) + \delta x(1 - \delta y)I(m + 1, n) + (1 - \delta x)\delta yI(m, n + 1) + \delta x\delta yI(m + 1, n + 1)$$

Avec  $\delta x, \delta y \in [0, 1]$

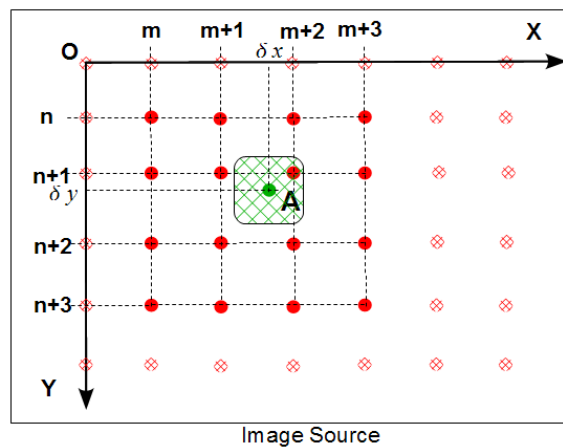
*Équation 5 Interpolation bilinéaire d'un point*



*Figure 5 Interpolation bilinéaire*

### 6.4.5.3 INTERPOLATION BICUBIQUE

L'interpolation bicubique va au-delà de la méthode bilinéaire, en prenant en compte les 16 pixels les plus proches du point à interpoler. L'idée est d'ajuster un modèle polynomial sur les 16 niveaux de gris de l'image source, puis d'en déduire le niveau du point interpolé par calcul de la valeur prise par le modèle. Le voisinage est centré autour du point A (même technique de positionnement par arrondi de la valeur des coordonnées (x, y) de A).



*Figure 6 Domaine de modélisation de l'interpolateur bicubique*

On remarque que le modèle bicubique possède 16 points de contrôle.

La résolution donne l'expression des coefficients ( $\alpha_{pq}$ ) en fonction des niveaux de gris de l'image source. La valeur prise par le point A sera donc :

$$NG(A) = \sum_{p=0}^{p=3} \sum_{q=0}^{q=3} \alpha_{pq} \delta x^p \delta y^q$$

# Diallo Alpha Oumar Binta – Rectification Plane d’Image (TER)

## Équation 6 Interpolation bicubique d’un point

*L’interpolation bicubique donne de meilleurs résultats que les méthodes précédentes; avec un bon rapport temps de calcul/qualité de rendu.*

### 6.4.5.4 EXEMPLE DE RÉSULTAT TROUVÉ



Figure 7 Exemple de rectification plane d’une image selon les deux choix possibles

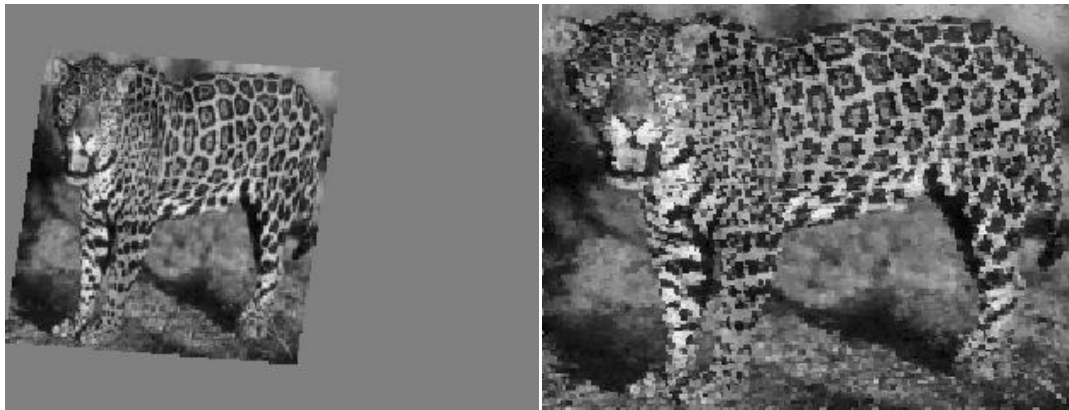


Figure 8 Exemple de rectification selon la taille de l’objet

## 6.5 L'INTERFACE GRAPHIQUE

La mise en place d'une charte graphique permet une cohérence au niveau interface, ainsi chaque élément trouve sa définition et son utilisation dans sa manière d'être représenté. Notre langage de programmation met à notre disposition un bon nombre d'outils permettant une plus grande simplicité d'utilisation, et il convient bien évidemment de les utiliser à bon escient.

### 6.5.1 ÉLÉMENTS GRAPHIQUES UTILISÉES

L'étude mise en place nous a conduits à appréhender les éléments graphiques suivants :

- La barre de menu, pour contrôler entièrement l'application.
- Les boutons standards, pour l'aiguillage vers une partie quelconque de l'interface graphique.
- Les boutons radios et combo box, pour une sélection unique dans une liste de choix.
- Les zones de textes, pour la saisie d'informations au clavier.
- Les onglets permettant l'affichage de l'image.
- Les boutons à cocher pour la sélection d'un ensemble de choix dans une liste.

Après avoir défini l'ensemble de besoins en ce qui concerne les outils graphiques, il convient de mettre en forme la fenêtre principale. Aux vues des besoins fonctionnels, nous proposerons une interface graphique principale définie par quatre zones d'informations ou d'interactions.

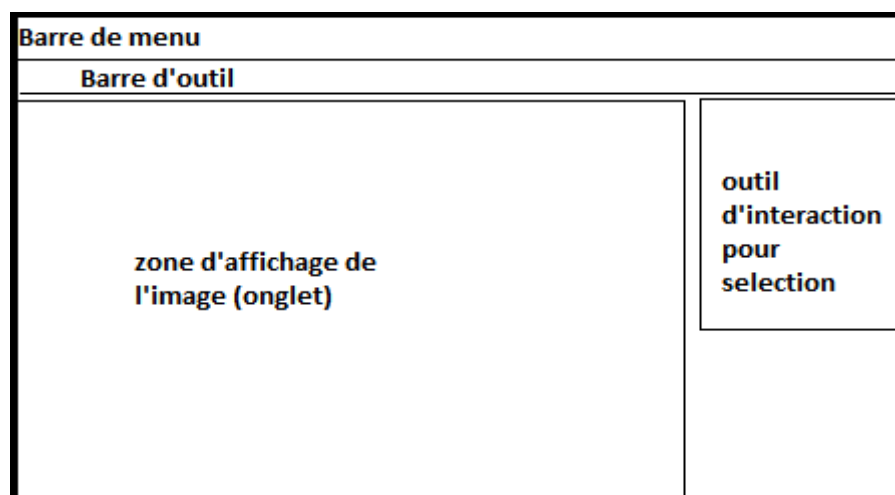


Figure 9 Proposition d'interface graphique

L'image est affichée par un JPanel qui est, lui-même contenu dans un panneau. Les boutons du panneau à outils, de la barre d'outils et certains éléments du menu ne sont pas accessibles tant qu'une image n'est pas ouverte.

## 6.5.2 EXPLICATIONS DES ZONES DE L'INTERFACE GRAPHIQUE

La barre de menu, située en haut de la fenêtre principale, constitue l'élément majeur permettant de contrôler l'ensemble des fonctionnalités de l'application. Ainsi, cette barre permet d'interagir avec l'image.

Une barre d'outils est contenant des boutons qui permettent d'exécuter des fonctions de l'application sans chercher dans les menus.

La zone principale de l'interface graphique se compose d'un onglet contenant l'image en cours de modification.

La quatrième et dernière zone contient les outils d'interactions entre l'utilisateur et l'image par l'intermédiaire de l'interface graphique pour la sélection précise de points (création, redimensionnement, déplacement et suppression de figure).

Ainsi nous proposons à l'utilisateur l'interface graphique suivante :

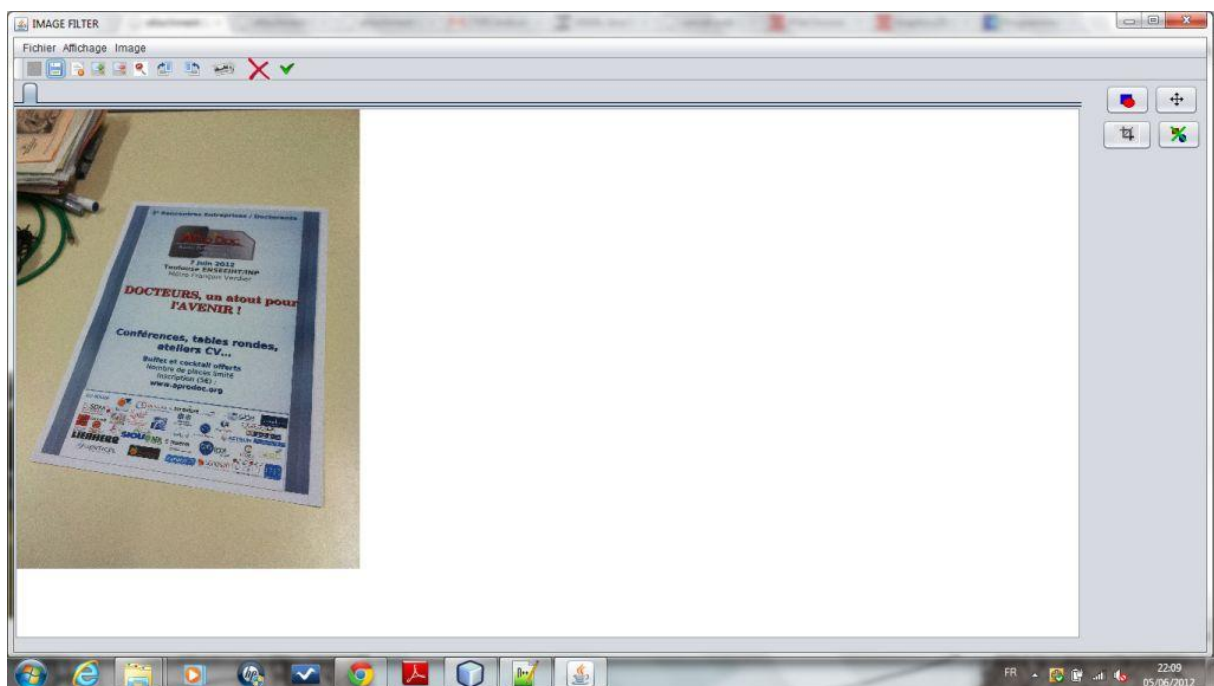


Figure 10 Interface graphique de l'application

# Diallo Alpha Oumar Binta – Rectification Plane d'Image (TER)

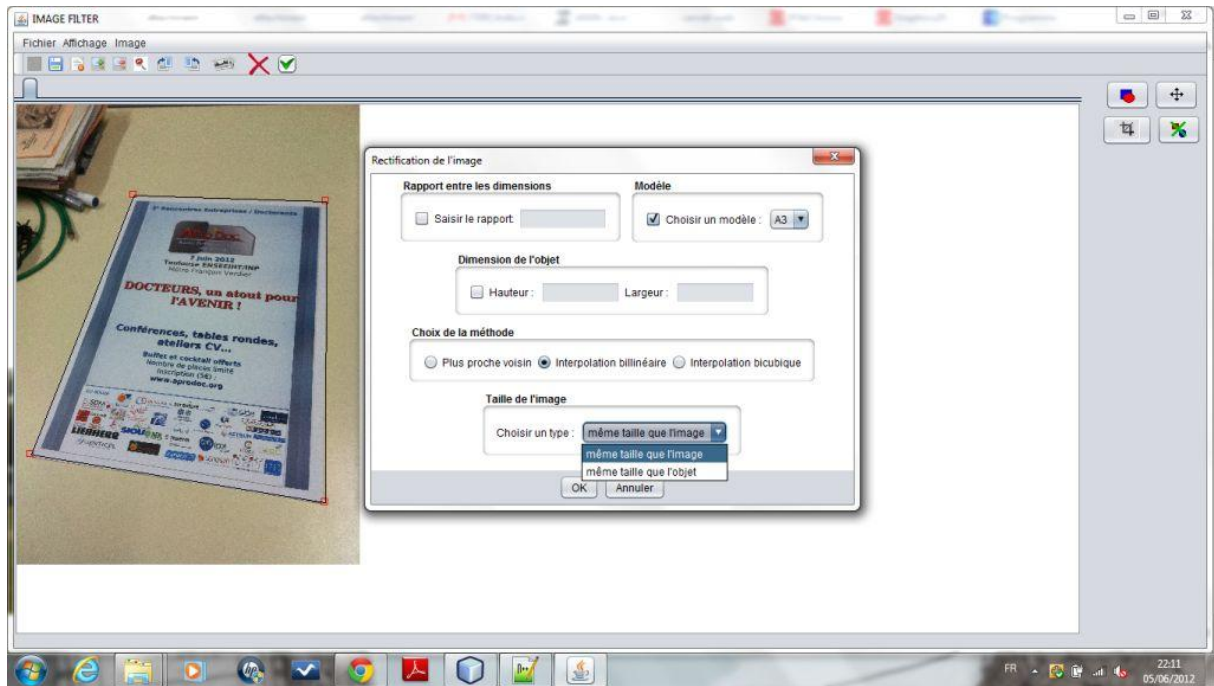


Figure 11 Paramétrage de la rectification d'une image



## 6.5.3 EXEMPLE DE BOITES DE DIALOGUES UTILISÉES

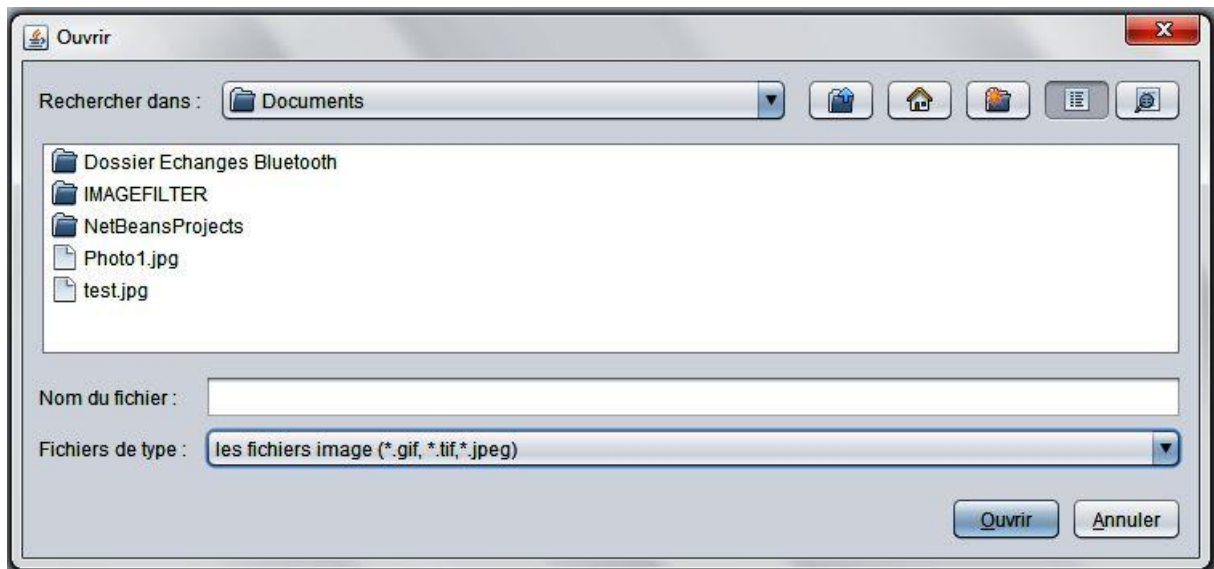


Figure 12 Boite de dialogue pour ouvrir un fichier

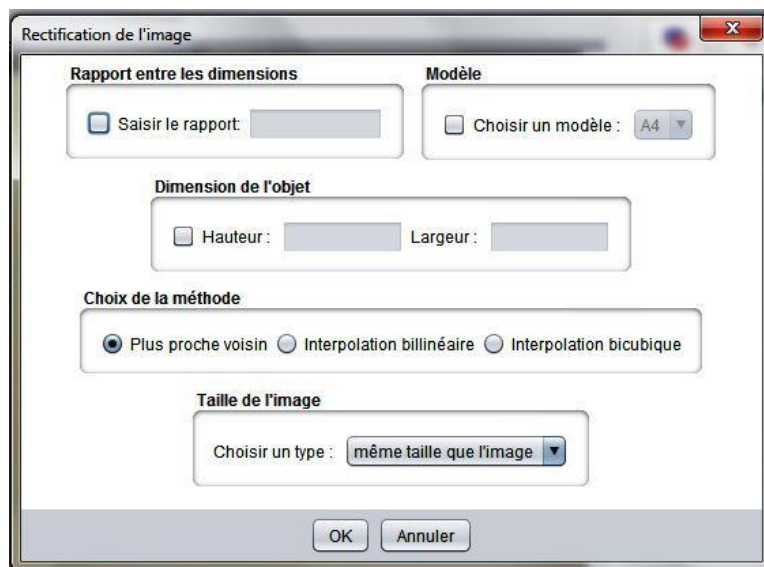


Figure 13 Boite de dialogue pour entrer les informations nécessaires pour effectuer la transformation

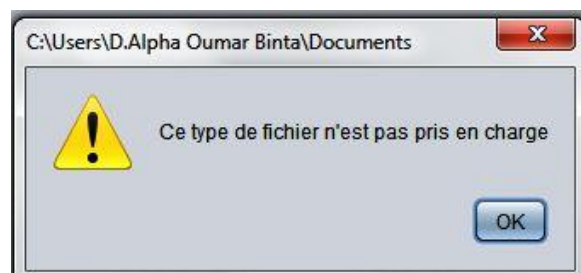


Figure 14 Exemple de message d'erreur



## 7 CONCLUSION

Après avoir présenté le sujet de ce TER, nous venons de voir comment, conceptuellement parlant ainsi que du point de vue du développement j'ai répondu aux attentes. Les objectifs d'un tel TER sont multiples : Il faut tirer une double expérience (communication omniprésente avec le responsable du sujet et acquisition de nouvelles connaissances dans le domaine ciblé par le TER).

Une perspective d'évolution de l'application serait de :

1. De généraliser la transformation géométrique aux images couleur.
2. De perfectionner l'outil de sélection dans l'image (effet de loupe).
3. Implémenter d'autres méthodes de rectification n'utilisant pas un rectangle de la scène.
4. De créer un outil de détection automatique de point d'intérêt.
5. De permettre le détournement automatique d'objet de la scène.

La durée de ce TER de sept semaines, se révèle être une expérience très bénéfique. J'ai réellement pu dépasser le stade de « formation ». Ensuite, le TER devient une vraie expérience professionnelle puisque j'ai été amené à travailler de manière autonome. Il m'a permis d'appliquer les connaissances qui m'ont été inculquées au cours de ces trois années de Licence ainsi que de m'initier à la recherche dans une optique éventuelle de poursuite d'étude dans le domaine du traitement d'image. J'ai été confronté à de nombreux problèmes et dans la plupart des cas j'ai pu trouver une solution alternative afin de les résoudre. Ce projet a été l'occasion pour moi de découvrir et d'utiliser des outils dont je n'avais pas la moindre idée de leurs existences notamment en matière de traitement d'image. Il est rassurant, en sortant de l'université, de voir que la formation reçue est pertinente et nous offre une grande autonomie.

Ce TER a répondu à mes attentes aussi bien au niveau professionnel qu'en termes de compétences relatives au savoir-faire nécessaire pour toute autre activité relationnelle. Il est mon premier pas vers un apprentissage de la réalisation complète d'une application.

## 8 BIBLIOGRAPHIE

Dans le cadre du stage ou TER du cursus licence informatique à l’université Paul Sabatier, il m’a été demandé de développer une application permettant la rectification plane d’une image. Afin de mieux aborder le sujet, j’ai dû faire appel à certains documents.

### Site Internet

Exif info (consulté le 17 Avril 2012) [En ligne]

Adresse URL: <http://www.siteduzero.com/tutoriel-3-31523-exif-lire-les-infos-d-une-image.html>

Adresse URL : <http://www.drewnoakes.com/code/exif/>

Tutorial Java (consulté le 20 Avril 2012) [En ligne]

Adresse URL: <http://www.siteduzero.com/tutoriel-3-10601-apprenez-a-programmer-en-java.html>

Planar Rectification (consulté le 26 Avril 2012) [En ligne]

Adresse URL: <http://www.robots.ox.ac.uk/~dl/planar.html>

### Document

Homographie et rectification plane d’image ([Alain Crouzil](#))