

## Registration of spatialised data

---

*Deliverable number : D13*

*Nature : P*

*Contractual Date of Delivery : April 2000*

*Task WP2.3 : Registration of spatialised data*

*Authors :*

*Alex Corenthin*

*Mouhamed Tidiane Seck*

*Aimé Cyriaque Ngatse*

*Institute :ESP /UCAD*

*Address :BP 15915 Dakar-Fann*

*Senegal*

*Email : [corenthin@esp.simes.sn](mailto:corenthin@esp.simes.sn)*

### **ABSTRACT**

The present document describes the techniques developed within the consortium for the registration of spatialised data and the mathematical grounds of these techniques. The application of these techniques, selected as the most relevant for SIMES WISE-DEV's needs, is the mosaicing of aerial images of the Niger river Central Delta.

The document details the algorithms used to automatically compute the relevant homographies for the registration of original adjacent images and to apply these homographies to obtain the image mosaic requested on line by the user, according to the scale and center specified.

The main advantage of this method is its robustness, and particularly its capacity to handle non calibrated images (i.e. the orientation and altitude of the camera used during the shooting of the aerial photographs does not need to be known).

This document constitutes the deliverable D13.

Key Words : image registration ; image mosaicing ; non calibrated images

<b><u>INTRODUCTION</u></b> .....	<b>3</b>
<b><u>APPROCHE INTUITIVE ET PRINCIPES</u></b> .....	<b>3</b>
<b><u>METHODE DE MISE EN CORRESPONDANCE DE GRAPHES</u></b> .....	<b>5</b>
<u>DÉFINITION</u> .....	5
<u>ALGORITHME DE « GRAPH MATCHING »</u> .....	5
<u>ALGORITHME D'APPROXIMATION POLYGONALE</u> .....	6
<u>ALGORITHME DE CALCUL DE LA TRANSFORMATION</u> .....	6
<b><u>PROGRAMMATION AUTOMATIQUE DES HOMOGRAPHIES</u></b> .....	<b>7</b>
<u>NOTIONS DE TRANSFORMATIONS PROJECTIVES</u> .....	7
<u>Notations et définitions</u> .....	7
<u>Notion de Matrice</u> .....	7
<u>Définition de l'image</u> .....	8
<u>Définition des homographies</u> .....	9
<u>DESCRIPTIONS DES TRANSFORMATIONS ET DE LEURS PROPRIÉTÉS</u> .....	10
<u>Isométries</u> .....	10
<u>Transformations de similarités</u> .....	10
<u>Transformations affines</u> .....	10
<u>Transformations projectives</u> .....	11
<u>IMPLÉMENTATION</u> .....	11
<u>ALGORITHME DE TRANSFORMATION LINÉAIRE DIRECTE (DLT=DIRECT LINEAR</u> <u>TRANSFORMATION)</u> .....	11
<u>Algorithme du DLT de base</u> .....	12
<u>DLT Normalisé</u> .....	12
<u>AUTRES MÉTHODES PROBABILISTES</u> .....	12
<u>Algorithme du RANSAC</u> .....	12
<u>TECHNIQUES DE SUPER-RÉSOLUTION</u> .....	13
<u>ALGORITHME DU MOSAÏCAGE</u> .....	14
<b><u>RESULTATS ET MISE EN OEUVRE DE L'INTERFACE WEB</u></b> .....	<b>16</b>
<u>I-PRÉPARATION DE LA MOSAÏQUE</u> .....	16
<u>II- DÉTERMINATION AUTOMATIQUE DES HOMOGRAPHIES</u> .....	17
<u>III -MOSAÏQUE</u> .....	18
<b><u>INTERFACE AVEC LE WEB</u></b> .....	<b>21</b>
<b><u>MISE EN LIGNE D'OUTILS DE TRAITEMENT D'IMAGE</u></b> .....	<b>24</b>
▪ <u>REHAUSSEMENT DE CONTRASTE</u> .....	25
▪ <u>AJOUT DE BRUIT</u> .....	25
▪ <u>SUPPRESSION DE BRUIT PAR LE FILTRE MÉDIAN</u> .....	25
▪ <u>MISE EN ÉVIDENCE DE FORME MORPHOLOGIQUE</u> .....	25
▪ <u>SEGMENTATION</u> .....	25
<b><u>BIBLIOGRAPHIE</u></b> .....	<b>25</b>

## **INTRODUCTION**

Parmi les applications des techniques de recalage d'images (image registration) étudiées dans le cadre du projet SIMES WISE-DEV, celle qui a été retenue comme la plus intéressante pour les opérations pilotes du projet est le mosaïchage automatique d'images non calibrées (c'est-à-dire prises par une caméra dont la position et l'orientation sont inconnues). Cette application utilise des techniques algorithmiques de pointe mises au point récemment par les membres du consortium (notamment l'Université d'Oxford et l'INRIA), basées sur le calcul automatique de l'homographie entre deux images présentant un recouvrement partiel.

Cette approche permet de calculer à la demande une image « mosaïque » à partir de plusieurs images originales (ici des images aériennes du Delta Central du fleuve Niger, prises par l'IGN à une altitude d'environ 4.000 mètres) qui se recouvrent partiellement (environ 20% de recouvrement entre deux images adjacentes). L'image mosaïque est calculée en fonction des paramètres fournis par l'utilisateur (centre et échelle) à partir d'une banque d'images stockées sur l'un des sites en réseau du projet (ici le site SIMES WISE-DEV de l'ESP à Dakar). La spécification des paramètres se fait de manière intuitive par navigation géographique à partir d'une des trois mosaïques globales réalisées pour chacune des trois zones d'étude (Batamani, Korientzé et Walado) de l'Observatoire de la pêche dans le Delta Central du Niger (opération pilote n°1 du projet) : l'utilisateur désigne simplement le centre de la région particulière qui l'intéresse et choisit éventuellement un facteur de zoom (rapprochement, éloignement, ou inchangé).

Le présent document passe en revue les méthodes les plus éprouvées pour arriver à obtenir de manière automatique une image mosaïque à partir d'une séquence d'images. La rédaction intègre un objectif pédagogique à destination des étudiants des institutions partenaires : elle incorpore donc volontairement de nombreux rappels mathématiques de base.

## **APPROCHE INTUITIVE ET PRINCIPES**

La reconstitution manuelle d'une image mosaïque à l'aide d'un logiciel de dessin tel que Photoshop ou Gimp, ou sans outil logiciel, consisterait à placer les différentes images sur des calques que l'on superposerait à partir des zones communes. Intuitivement, on procède au choix d'une référence commune à deux images voisines et on détermine les translations (ou plus généralement les homographies) permettant de mettre en correspondance les images.

Les deux méthodes d'automatisation du mosaïchage que nous avons choisi de présenter reposent sur cette démarche intuitive. Cependant, elles utilisent des fonctions plus riches que les simples translations, pour pouvoir répondre aux contraintes liées aux mouvements de la caméra embarquée qui peut effectuer non seulement des mouvements de translation, mais également des zooms (qui peuvent aussi résulter de variations d'altitude de la caméra), ainsi que des rotations autour de son axe.

Nous voyons donc que le problème principal pour la réalisation de la mosaïque est la recherche des zones communes à superposer, et par conséquent de réaliser le recalage de ces images avant la superposition.

Dans la première méthode, ces zones communes sont obtenues en utilisant des algorithmes de recherche opérationnelle sur la mise en correspondance de graphes (« graph matching »), les images sont d'abord segmentées puis modélisées sous forme de segments de droite qui conduisent à la génération pour chaque image d'un graphe dont les sommets correspondent aux segments de droite et les arcs aux angles entre ceux-ci. Ces graphes sont mis en correspondance pour déterminer les zones communes qui vont permettre alors de calculer la transformation entre les deux images.

Dans la deuxième méthode, ces zones communes sont obtenues en utilisant une estimation automatique des homographies entre les deux images en utilisant des notions probabilistes dans la programmation automatique des homographies.

Les séquences d'images que nous manipulons dans le cadre du projet, ayant d'importantes zones communes avec leurs voisines, les transformations précédentes ne suffisent pas à leur traitement en temps réel, il faut pour lever cette contrainte, diminuer la résolution des images, ce qui nous permet d'obtenir pour une même séquence plusieurs mosaïques à des niveaux de résolutions différents, dont la plus haute est celle de l'image originale. C'est l'application de la technique dite de super-résolution.

Le document détaille les algorithmes utilisés par les deux méthodes de mosaïcage (« graph matching », programmation automatique des homographies) mais également aussi la technique de super-résolution utilisée. Il décrit également l'algorithme de mosaïcage mis en œuvre sur la base des résultats obtenus et présente les résultats sur les séquences d'images de nos zones d'intérêt ainsi que l'interface mise en place pour visualiser et présenter ces résultats dans le cadre du projet SIMES WISE-DEV.

## METHODE DE MISE EN CORRESPONDANCE DE GRAPHES

Dans cette partie nous allons décrire :

- la méthode de mise en correspondance de graphes plus connue sous l'appellation d'algorithme de « graph matching »
- l'algorithme d'approximation polygonale pour l'obtention des segments de droite
- l'algorithme permettant de calculer la transformations une fois les régions communes à la séquence traitée mises en correspondance.

### Définition

Un graphe est défini comme un 4-tuple :  $G(V,E, \mu, \gamma)$

- $V$  représente l'ensemble des sommets
- $E$  représente l'ensemble des arcs ou contours :  $E \subseteq V \times V$
- $\mu$  est une fonction  $V \rightarrow L_V$  des noms des sommets
- $\gamma$  est une fonction  $E \rightarrow L_E$  des noms des arcs en supposant que les arcs sont directs

### Algorithme de « graph matching »

L'idée est simple, étant donné deux graphes, il s'agit de déterminer l'ensemble des points de la première image qui se trouvent dans la seconde image, chaque image étant représentée sous forme de graphe.

Soit deux graphes  $G_1(V_1, E_1, \mu_1, \gamma_1)$  et  $G_2(V_2, E_2, \mu_2, \gamma_2)$  :

où  $V_1 = \{v_1, v_2, v_3, \dots, v_{n1}\}$  et  $V_2 = \{w_1, w_2, w_3, \dots, w_{n2}\}$   
sont les ensembles des sommets de  $G_1$  et  $G_2$

```
Initialiser une Liste OPEN := NULL et le Seuil := +∞
⊗  $V_2$  calculer P et déterminer le coût correspondant tel que  $P = [(v_1, w)]$ 
et OPEN := OPEN ∪ {P}.
  if OPEN := NULL ALORS exit
  endif
tant que Non Complete matching alors
  faire
  Pmin est le P de minimum coût dans OPEN
  if C(Pmin) > seuil alors exit
  Seuil := C(Pmin)
  endif
  Pmin = [(x1, y1), ..., (xk, yk)] et  $V_2^P = \{w / (v, w) \in P, w \in V_2\}$ 
  If ( $k < |V_1|$ ) alors
     $w \in V_2 - V_2^P$  { } Pnew = [(x1, y1), ..., (xk, yk), (vk+1, w)]
    OPEN = OPEN ∪ {Pnew} else if ( $V_2 - V_2^P \neq \text{NULL}$ ) alors
      Pnew = [(x1, y1), ..., (xk, yk), (ε,  $\tilde{y}$ )] avec  $\tilde{y} \in V_2 - V_2^P$ 
      OPEN = OPEN ∪ {Pnew}
    else exit
```

**Complete matching :** La mise en correspondance (« graph matching ») est complète lorsque tous les points communs sont déterminés. Les points qui sont calculés avec des valeurs supérieures à un seuil donné sont les points de correspondance. Donc à l'issue de ce programme, on obtient une liste P contenant les points de correspondance avec l'erreur estimée dépendant du seuil choisi.

### Algorithme d'approximation Polygonale

Nous utilisons ici un module permettant d'obtenir une image segmentée présentant les composantes connexes des images initiales. Pour un composant connexe donné, les points des contours sont enregistrés dans un tableau ; l'approximation polygonale permet d'obtenir l'ensemble des sommets de cette courbe stockés dans un tableau d'entier. La procédure retourne également le nombre de points retenus dans l'approximation polygonale.

La méthode consiste pour chaque segment, à rechercher le point le plus éloigné par rapport à un critère de distance. Si ce point existe, il est considéré comme élément du nouveau segment à estimer et est introduit dans l'ensemble des sommets de l'approximation. La procédure est récursive et ne s'arrête que si aucun point ne répond plus au critère.

Grâce à cette approximation, nous générons des graphes dont les lignes sont positionnées par les sommets consécutifs de l'approximation et dont la longueur est déterminée par la distance entre ces sommets. Les arcs sont estimés par l'angle entre deux segments de droites consécutifs.

Cet algorithme permet de générer un graphe pour chaque image constituant la séquence.

### Algorithme de calcul de la transformation

Pour obtenir l'ensemble des points de correspondance entre deux images dont on a déterminé les graphes par approximation polygonale, on applique l'algorithme suivant :

Soit S l'ensemble des points de l'image G1 et S' l'ensemble des points de l'image G2 à mettre en correspondance :

On calcule la translation en cherchant la moyenne des points S et S'

La translation est alors ( $t_x = S_x - S_x'$  et  $t_y = S_y - S_y'$ )

On calcule la variance par :

$$\text{Var1} = (x - S_x)^2 + (Y - S_y)^2$$

$$\text{Var2} = (x' - S_x')^2 + (Y' - S_y')^2$$

Le facteur d'écart est  $\sigma^2 \text{Var1} = \text{Var2}$

La direction des axes relative à S

$$m_{11} = \frac{(x - S_x)(y - S_y)}{\text{nombre de valeur}}$$

$$m_{20} = \frac{(x - S_x)}{\text{nombre de valeur}}$$

$$m_{02} = \frac{(y - S_y)}{\text{nombre de valeur}}$$

On fait de même pour S' en remplaçant respectivement m11, m20, et m02 par m'11, m'20, et m'02 ; et x,y,Sx,Sy par x',y',Sx',Sy'.

Par la suite on calcule les variables  $\alpha$  et  $\alpha'$  :

$$2\alpha = \tan(m_{11}/(m_{20}-m_{02}))$$

$$2\alpha' = \tan(m'_{11}/(m'_{20}-m'_{02}))$$

la rotation à effectuer aura pour angle  $(\alpha' - \alpha)$

La transformation ou l'homographie cherchée est composée de la translation calculée multipliée par le facteur  $\sigma$  suivie de la rotation.

## PROGRAMMATION AUTOMATIQUE DES HOMOGRAPHIES

### Notions de transformations projectives

#### Notations et définitions

Nous adopterons les notations suivantes :

P est un plan vectoriel ; dans ce plan, un point est représenté par ses deux coordonnées (x,y), le couple (x, y) est un vecteur. Soit X un vecteur ligne, alors sa transposée notée  $X^T$  est un vecteur colonne.

#### Notion de Matrice

Une matrice est un tableau qui permet de représenter une application linéaire entre deux espaces vectoriels de dimensions finies.

En effet, si A est une application linéaire de E (de dimension n et de base  $(e_1, e_2, \dots, e_n)$ ) dans F (de dimension m et de base  $(f_1, f_2, \dots, f_m)$ ), alors nous notons  $A(e_j) = \sum_i^m a_{i,j} f_i$  soit

$$A(x) = A\left(\sum_j^n x_j e_j\right) = \sum_j^n x_j A(e_j) = \sum_i^m \left(\sum_j^n a_{i,j} x_j\right) f_i$$

avec  $x \in E, x = \sum_{j=1}^n x_j e_j$  alors  $x_i$  et  $a_{i,j}$  élément de C ou R

La matrice A représente l'application linéaire A sous la forme d'un tableau 2D d'élément courant. On remarquera que la matrice dépend du choix des bases de E et F. Dans la suite on verra que de très nombreux problèmes se ramènent à la résolution d'un système linéaire, en particulier dans le domaine des images.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{bmatrix}$$

Un système algébrique de n équations linéaires, à n inconnues,  $A \cdot X = B$  a en effet une signification différente suivant que le déterminant de la matrice A est nul (système singulier) ou non (système régulier). Par ailleurs, un système homogène (le second membre B est formé de composantes  $b_i$  toutes égales à zéro) doit être traité différemment d'un système inhomogène (l'une au moins des composantes  $b_i$  est différente de zéro).

Si le système est régulier (le déterminant de A est différent de zéro), deux cas doivent être envisagés :

- i) le système est inhomogène ( $B \neq 0$ )  
Il existe une et une seule solution X.
- ii) le système est homogène  
Il existe une seule solution, trivialement nulle :  
 $X = 0$ , ou  $x_i = 0$ , pour  $i = 1, 2, \dots, n$ .

Si le système est singulier (le déterminant de A est nul), on doit également envisager les cas où :

- i) le système est inhomogène ( $B \neq 0$ ) : il n'existe aucune solution X qui vérifie le système.
- ii) le système est homogène : il existe une ou plusieurs solutions complexes normalisées.

Une matrice triangulaire supérieure est une matrice A telle que  $a_{i,j} = 0$  si  $i < j$ . Elle est dite triangulaire inférieure si  $a_{i,j} = 0$  si  $i > j$ . Une matrice est diagonale si  $a_{i,j} = 0$  si  $i \neq j$ . L'intérêt des matrices diagonales ou triangulaires est qu'elles sont facilement inversibles (lorsqu'elles sont carrées).

Les matrices carrées sont les matrices  $n \times n$ . Ce sont les seules qui aient une chance d'être inversibles. Une condition nécessaire et suffisante d'inversibilité est que le déterminant de la matrice soit non nul. Cette condition n'est jamais utilisée numériquement pour des grandes matrices car l'évaluation d'un déterminant est numériquement coûteuse et potentiellement instable. De plus, le nombre réel zéro n'a pas toujours grand sens en machine après de nombreuses opérations. Deux matrices A et B sont dites semblables lorsqu'il existe une matrice inversible S telle que  $A = SBS^{-1}$ .

### Définition de l'image

Un pixel (contraction de "picture élément") est le nom associé à une unité de base de l'image qui correspond à la surface minimale. Un pixel est caractérisé par sa position (coordonnées spatiales) et sa valeur (i.e. son niveau de gris).

Une image est une matrice de points appelés pixels. L'information dont elle est le support est caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).

$f : [0, L-1] \times [0, C-1] \rightarrow [0, M]^p$  définit une image de  $L$  lignes et  $C$  colonnes dont l'information portée est définie dans un espace à  $p$  dimensions.

Si  $f$  est une image binaire, alors  $(p, M) = (1, 1)$

Si  $f$  est une image en niveaux de gris, alors  $p = 1$  et le plus souvent  $M = 255$

Si  $f$  est une image couleur, alors  $p = 3$  et le plus souvent  $M = 255$ .

Tout pixel d'une image est caractérisé par un couple de coordonnées  $(x, y)$ .

### Définition 1

Soit une ligne  $\mathbf{a}x^2 + \mathbf{b}x + \mathbf{c}$  où  $a, b, c$ , sont choisis pour différencier les lignes, une ligne est représentée alors par un vecteur  $(a, b, c)^T$ . Dans un espace à trois dimensions, le point  $x = (x, y)^T$  appartenant à la droite peut être écrit sous la forme  $(x, y, 1) \times (a, b, c)^T = 0$ .

Soit  $L = (a, b, c)^T$  et  $L' = (a', b', c')^T$  deux lignes, alors leur intersection est un point déterminé par l'équation  $X = L * L'$  ou  $*$  est le produit vectoriel des deux lignes.

### Définition 2

Une transformation projective est une carte inversible de  $P^2$  dans lui-même telle que trois points  $x_1, x_2, x_3$  sont liés sur une ligne si et seulement si  $h(x_1), h(x_2), h(x_3)$  le sont.

Preuve : Soit  $x_1, x_2, x_3$  trois points liés sur une ligne  $L$ , alors  $L^T x_i = 0$  pour tous  $i$ . entier.

Soit  $H$  une matrice  $3 \times 3$ , non singulière, on vérifie que  $L^T H^{-1} H x_i = 0$ , donc tous les points  $H x_i$  sont liés par  $H^{-1} L$ , la colinéarité est donc préservée par la transformation.

### **Définition des homographies**

Une transformation projective plane (homographie) est une transformation linéaire d'un vecteur homogène représentée par une matrice  $3 \times 3$  non singulière où brièvement  $x' = Hx$

Soit  $\begin{matrix} x'_1 \\ x'_2 \\ x'_3 \end{matrix} = \begin{matrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{matrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix}$  la transformation projective peut être écrite sous la forme

$$\begin{aligned} x'_1 &= \frac{x_1}{x_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' &= \frac{x_2}{x_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{aligned}$$

donc chaque point de correspondance génère deux équations pour les éléments de  $H$  qui sont

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

## Descriptions des transformations et de leurs propriétés

### Isométries

Les isométries sont les transformations planes de  $P^2$  qui préservent la distance euclidienne :

une isométrie est représentée par 
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \varepsilon \cos\theta & -\sin\theta & t_x \\ \varepsilon \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$
 ou  $\varepsilon = \pm 1$  si  $\varepsilon = 1$  alors l'orientation

est préservée par la transformation si  $\varepsilon = -1$  alors l'isométrie inverse l'orientation, une transformation euclidienne plane peut être résumée sous la forme

$x' = H_E x = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} x$  ou  $R$  = une matrice de rotation  $2 \times 2$ ,  $t$  = la translation et  $0$  est le vecteur nulle.

Les invariants de cette transformation sont la longueur (distance entre les points), l'angle (angle entre les lignes et la surface). Les orientations sont préservées si le côté gauche de la matrice  $2 \times 2$  à un déterminant égal à 1.

### Transformations de similarités

Une similarité est une isométrie composée avec une isométrie échelon Dans le cas des transformations euclidiennes composées avec une échelle, la similarité est une matrice représentée par

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s \cos\theta & -s \sin\theta & t_x \\ s \sin\theta & s \cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \text{ soit } x' = H_s x = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix} x$$

le scalaire  $s$  représente l'échelle de l'isométrie.

Cette transformation préserve la forme et a 4 degrés de liberté, aussi le rapport des distances est invariant ainsi que celui des surfaces.

### Transformations affines

Une transformation affine ou affinité est une transformation linéaire non singulière suivie par une translation, elle est représentée par :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \text{ soit } x' = H_A x = \begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix} x \text{ avec } A \text{ la matrice } 2 \times 2 \text{ non singulière.}$$

La transformation affine a 6 degrés de liberté correspondant au 6 élément de la matrice, elle est comme une composition de deux transformations fondamentales normalement rotation et mise à l'échelle.

La matrice  $A$  peut être représentée sous la forme d'une décomposition en valeurs singulières dite SVD,

$A=UDV^t$  où U et V sont des matrices orthogonales , D est une matrice diagonale .

Une affinité préserve l'orientation ou l'inverse selon que le déterminant de A est positif ou non.

### Transformations projectives

Une transformation projective, généralement non singulière généralise une affinité .

Elle peut être représentée sous la forme :  $x'=H_p x = \frac{A}{V^t} x$  avec V une matrice 2x2, la matrice a 9 éléments non nuls, elle peut être déterminée par 4 points de correspondance, l'invariant est le rapport des 4 points colinéaires.

En résumé toutes les transformations du plan peuvent être exprimées sous forme décomposée

$$H=H_s H_A H_p = \begin{pmatrix} sR & t & A & t \\ 0^t & 1 & 0^t & 1 \end{pmatrix} \begin{pmatrix} A & t \\ V^t & V \end{pmatrix} = \begin{pmatrix} sR & t & k & 0 \\ 0^t & 1 & 0^t & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ V^t & V \end{pmatrix} = \begin{pmatrix} A & t \\ V^t & V \end{pmatrix}$$

$A=sRK+tV^t$  ou K est une matrice triangulaire de déterminant 1 et  $V \neq 0$  et s choisi positif.

Dans la suite, notre objectif est de chercher à estimer la matrice H dite transformation homographique.

### Implémentation

Il s'agit de trouver quelle partie d'une image A se retrouve dans une image B de référence. Pour ce faire, on considère l'ensemble des points de correspondances  $x_i \rightarrow x'_i$  entre A et B et on cherche une matrice 3x3 H tel que  $x'_i=Hx_i$  .

### Algorithme de transformation linéaire directe (DLT=Direct Linear Transformation)

Il est décrit par la résolution de l'équation  $X_i^* X_i = 0$  Cette forme permet d'avoir une solution linéaire.

Soit  $h^{j^t}$  le  $j^{i\text{ème}}$  vecteur ligne de la matrice H, alors nous pouvons écrire  $HX_i = \begin{pmatrix} h^{1^t} \\ h^{2^t} \\ h^{3^t} \end{pmatrix}$

En posant  $X'_i=(x_i, y_i, w_i)$  on obtient  $X_i^* H X_i = \begin{pmatrix} y_i h^{3^t} x_i - w_i h^{2^t} x_i \\ w_i h^{1^t} x_i - x_i h^{3^t} x_i \\ x_i h^{2^t} x_i - y_i h^{1^t} x_i \end{pmatrix}$  puisque  $h^{i^t} x_i = x_i^t h^i$  pour

$i=1,2,3$ .

Ce qui entraîne 
$$\begin{pmatrix} 0^t & -w_i^t x_i^t & y_i^t x_i^t \\ w_i^t x_i^t & 0^t & x_i^t x_i^t \\ -y_i^t x_i^t & x_i^t x_i^t & 0^t \end{pmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (1)$$
 cette équation est de la forme  $A_i h = 0$

Ou  $A_i$  est une matrice  $3 \times 9$  et  $h = \begin{matrix} h^1 \\ h^2 \\ h^3 \end{matrix}$  soit  $H = \begin{matrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{matrix}$  (2)

c'est une équation linéaire avec  $h$  inconnu. On montre que la résolution de (1), compte tenu du fait que deux des équations sont linéairement indépendantes, se résume à la résolution de l'équation

$$\begin{matrix} 0^\tau & -w'_i x_i^\tau & y'_i x_i^\tau & h^1 \\ w'_i x_i^\tau & 0^\tau & x'_i x_i^\tau & h^2 \\ & & & h^3 \end{matrix} = 0 \quad (3) \text{ ou } A_i \text{ est maintenant une matrice } 2 \times 9$$

L'algorithme proposé ici est une méthode pour la résolution de cette équation

### Algorithme du DLT de base

- i) Pour chaque point de correspondance  $x_i \quad x'_i$ , calculer la matrice  $A_i$  (3).
- ii) regrouper les  $A_i$  résultantes des  $n$  points de correspondances dans une matrice  $A$  de dimension  $2n \times n$ .
- iii) Obtenir la décomposition en valeur singulière de  $A$ .
- iv) La matrice  $H$  est alors déterminée par (2).

### DLT Normalisé

- i) Normalisation de  $X$  : Programmer une similarité  $T$  composée d'une translation et d'une mise à l'échelle qui fait correspondre chaque ensemble de points  $X_i$  à un nouvel ensemble de points  $\tilde{X}_i$  tel que le centre de gravité de ces points  $\tilde{X}_i$  est de coordonnées d'origines  $(0,0)^\tau$  et leur distance moyenne à l'origine est de  $\sqrt{2}$ .
- ii) Normalisation de  $X'$  : même chose que pour i).
- iii) Application de l'algorithme du DLT de base pour la correspondance des points  $\tilde{X}_i \quad \tilde{X}'_i$  pour obtenir l'homographie  $\tilde{H}$ .
- iv) Dénormalisation : on calcule  $H = T \tilde{H} T$ .

*L'algorithme du DLT utilisant la normalisation est préférée à l'algorithme de base.*

### Autres méthodes probabilistes

Ces méthodes, parmi lesquelles nous avons choisi l'algorithme d'estimation robuste (RANSAC), utilisent des règles probabilistes pour l'estimation des correspondances entre les images d'une séquence.

### Algorithme du RANSAC

*(RANdom SAMple Consensus, de Fischer et Bolles)*

## Objectif

Sur une image à deux dimensions, trouver la ligne qui minimise la somme des carrés des distances des points de correspondance sur  $t$  itérations. [in *Multiple View Geometry in Computer Vision*, R. Hartley & A Zisserman, Cambridge University Press, June 2000]

## Implémentation

Les données en entrée de l'algorithme sont des images et la sortie est une homographie qui contient un ensemble des points d'intérêts dans les correspondances.

La première étape de l'algorithme est d'identifier les points d'intérêts dans chaque image, car ces points sont suffisants pour la détermination d'une homographie stable.

## Algorithme

- i) Détermination des points d'intérêt dans chaque image.
- ii) Permutation des correspondances : Calculer l'ensemble des points mis en correspondance en se basant sur la proximité et la similarité des intensités de leurs voisins.
- iii) Démarrage de l'algorithme RANSAC.
- iv) Optimisation de l'estimation : ré-estimation de  $H$  pour toutes les classes de correspondances, classification des points de correspondance, minimalisation de la fonction de coût du maximum de probabilité en utilisant l'algorithme de Levenberg-Marguardt.
- v) D'autres points de mise en correspondance peuvent être déterminés en utilisant l'estimation  $H$  obtenu précédemment.

## Techniques de super-résolution

Cette technique consiste à diminuer la résolution des images initiales de la séquence par un facteur multiple de 2., par exemple, une façon simple de diviser par deux la résolution consiste à remplacer 4 pixels d'une image par un unique pixel, dont l'intensité est la moyenne des 4. Cependant, cela crée des discontinuités importantes au sein de l'image, pour y remédier, l'intensité du pixel résultat sera déterminé par moyenne des pixels voisins pondérés par la fonction de Gauss. Suivant la résolution souhaitée, on divise par  $2^n$  la résolution de l'image d'origine à traiter de façon à obtenir une représentation pyramidale de celle-ci.

Sur les  $n$  images composant la séquence, La méthode consiste à calculer tout d'abord les fonctions  $H_0$  permettant de superposer deux par deux, les images de plus basse résolution dans l'ordre de la mosaïque.

### 1<sup>ère</sup> phase : Ordonnement des images

Pour une image  $k$  de la séquence, on obtient  $n-1$  transformations  $H_{k,j}$  avec  $j < k$  par rapport aux autres images de la séquence. La transformation retenue  $H_k$  est celle qui minimise les critères de coût.

On applique sur cette base un algorithme de tri qui permet de déterminer l'ordre de la séquence des images.

### 2<sup>ème</sup> phase : Détermination des homothéties de résolution

On calcule la transformation  $H_0$  permettant de superposer les images de basse résolution puis on transforme l'image source de la résolution supérieure avec la fonction venant d'être calculée  $H_{0,1}=H_0$ , afin d'obtenir une image intermédiaire, cette dernière est maintenant suffisamment proche de l'image destinataire pour pouvoir calculer la fonction projective  $H_1$  permettant de les superposer aux niveaux suivants. On réitère le procédé sachant que l'image intermédiaire est calculée en appliquant  $H_{0,k}=H_{k-1} \circ H_{0,k-1}$  à l'image source.

On détermine les homographies entre toutes les  $n$  images et la première image  $0$  :  $H_{n,0}=H_{n,n-1} \cdot H_{i,i-1} \cdot H_{1,0}$  ;  $H_k=H_{k,0}$ .

Lorsque l'on arrive aux deux images initiales (si la pyramide de gauss a  $n$  étages), en composant  $H_n \circ H_{0,n-1}$  on obtient la projection de la superposition des deux images initiales.

### 3<sup>ème</sup> phase : Mosaïcage

Voir la description détaillée de l'algorithme, ci-dessous.

### Algorithme du MOSAÏCAGE

L'Algorithme est subdivisé en trois grandes parties :

Soit  $i$  variant de 1 jusqu'au nombre d'images de la séquence capables de reconstituer la scène.

Nous notons ici que  $(x, y, w, h)$  signifie que  $(x, y)$  sont les coordonnées de l'origine de l'image et  $(w, h)$  sont respectivement la longueur et la hauteur de l'image

- Nous procédons d'abord à la détermination des coordonnées de l'image mosaïque  $(x', y', w', h')$  en utilisant les homographies et les coordonnées  $(x_i, y_i, w_i, h_i)$  de l'ensemble des images composant la scène. Ces images sont reliées par les homographies  $X'_i = H X_i$ . Pour chaque image de coordonnées

$(x_i, y_i, w_i, h_i)$  on détermine en utilisant les homographies leurs transformées  $(x'_i, y'_i, w'_i, h'_i)$  en suite on détermine le plus petit des  $x'_i$  et  $y'_i$  ce qui donne le couple  $x', y'$  et on fait de même pour déterminer le maximum des  $w'_i$  et  $h'_i$  ce qui donne  $(w', h')$ .

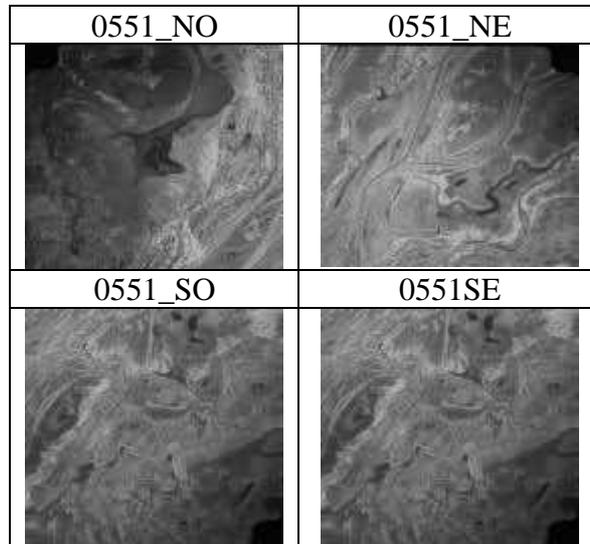
- Ensuite, il faut choisir l'image du début de mosaïque, écrire les points transformés de cette image sur la matrice de l'image mosaïque en utilisant l'homographie correspondante. Les autres images sont alignées avec les approximations de concaténation déterminées par les homographies et la technique de super résolution, enfin, on applique le MLE sur l'image mosaïque pour la réduction des erreurs de recalages accumulés.

## RESULTATS ET MISE EN OEUVRE DE L'INTERFACE WEB

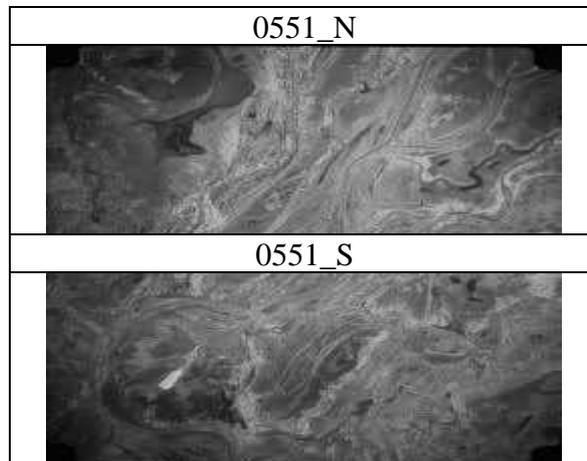
### I-Préparation de la mosaïque

Reconstruction de l'image disponible en 4 positions NE, NO, SE, SO

Par exemple l'image N°0551 se présente sous la forme :



En utilisant la commande Pnmcat obtient : avec l'option `-lr` ; on passe de 4 à 2 images



En utilisant l'option `-tb` , la fusion est terminée



On répète la procédure pour toutes les images disponibles. Ce que nous avons testé sur 43 images.

Enfin on utilise la commande **Pnmscale** pour créer les différents niveaux d'échelle ce qui achève l'étape de préparation

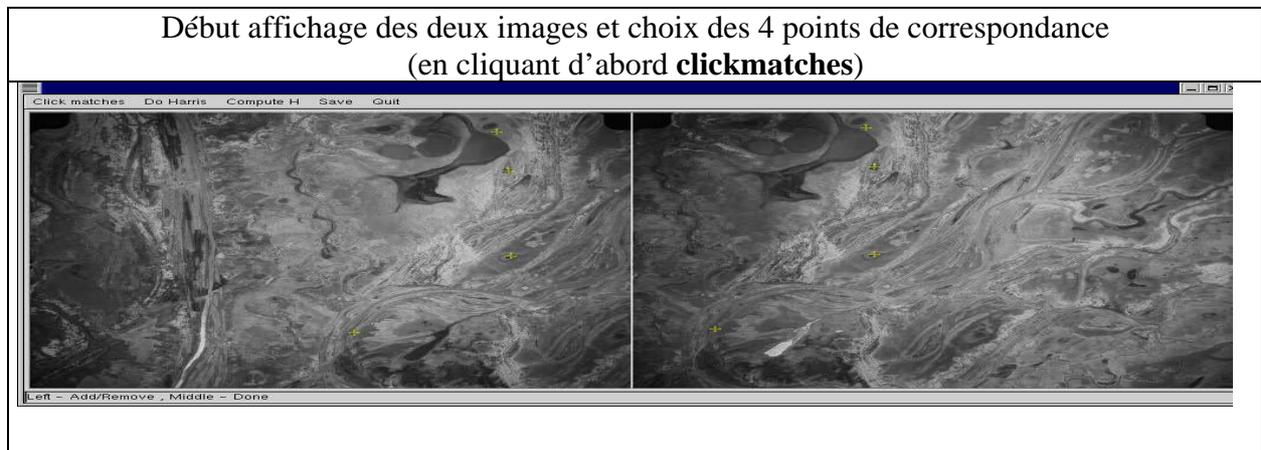
## II- Détermination automatique des homographies

L'objectif du programme est de déterminer l'homographie optimale qui permet le passage de l'image 1 à l'image 2

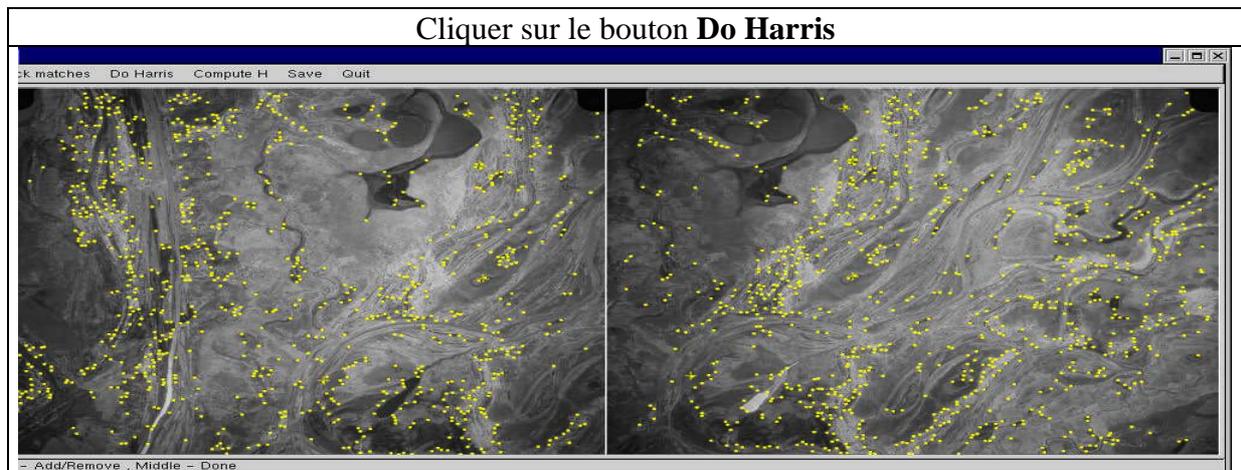
Voici un exemple de détermination automatique des homographies entre image 0550 et 0551

- Affichage des deux images et choix de 4 points de correspondance dans ces images

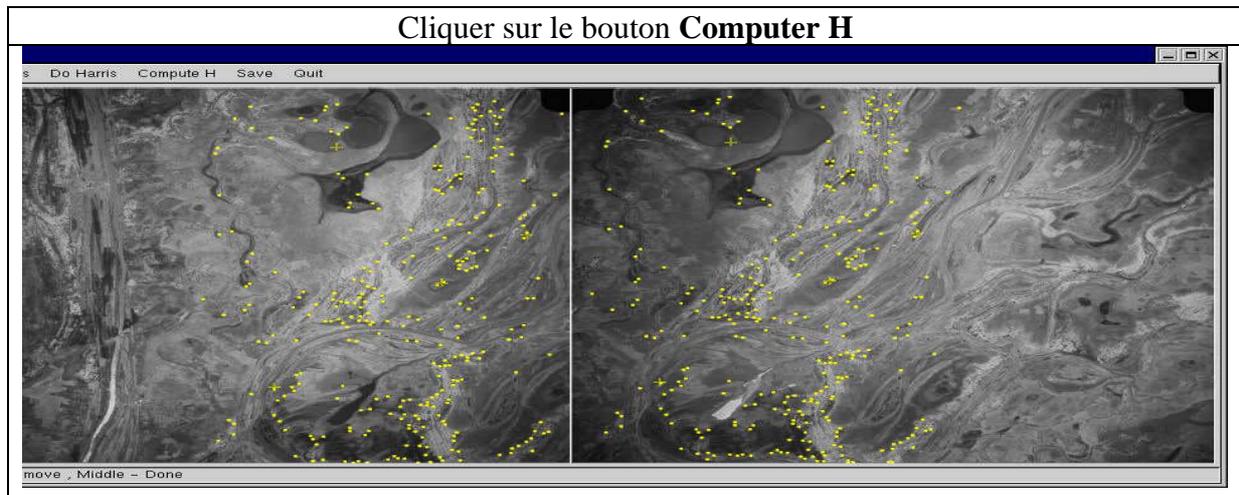
**Two\_view\_tool** 0551\_5.pgm 0551\_5.pgm



- Affichage des points de Harris pour préparer la génération des matches



- Le programme estime le nombre de points de correspondances possibles plus de 200 sont souhaitables avant de sauvegarder.



Puisse que nous avons obtenu plus 200 matches , nous pouvons sauvegarder le résultat. Le programme estime l’homographie qui convient le mieux lorsqu’on clique sur le bouton **save** ; si non on reprend l’opération en changeant les 4 points initiaux.

Nous avons obtenu pour ces deux images les matrices homographies optimale suivantes :

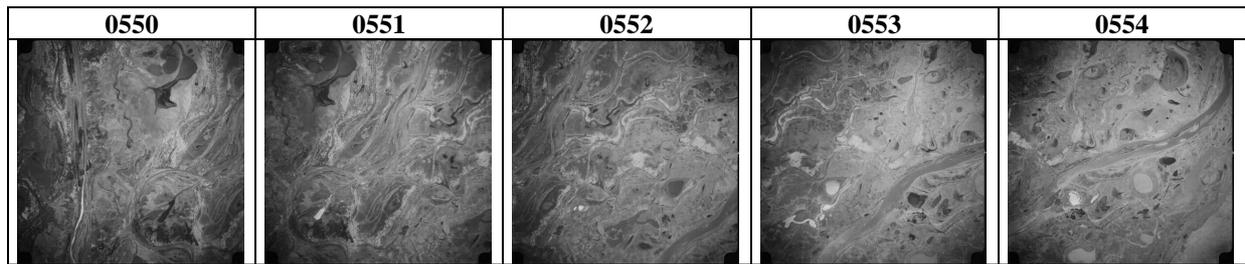
<b>Matrice Homographie conditionnée</b>	<b>Matrice d’Homographie</b>
1.00330217 -0.00257319 - 200.77106695	0.96932430 0.00803865 - 192.73878915
0.01043297 1.03594893 - 13.614280401	-0.02458207 0.97561546 8.0134738636
-0.00003705 0.00009895 1.0000000000	-0.00004905 -0.00001495 1.0000000000
=0550.0551.H	=0550.0551.cond_H

Cette étape de la détermination des homographies est répétée pour les différentes images que l’on veut superposer

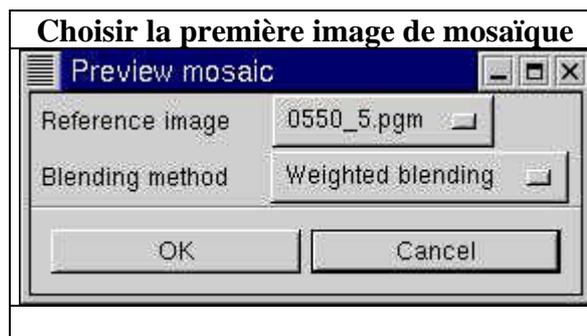
### **III -Mosaïque**

L’étape de la génération de l’image mosaïque est faite en utilisant le programme SIMES WISE-DEV\_mosaic, Ici le seul paramètre d’entrée est le fichier qui contient la liste des images à mosaïquer.

Voici les images de départ :



**SIMES WISE-DEV\_mosaic** listeA est le fichier qui contient les numéros des images





Nous signalons que l'image mosaïque est obtenue sous forme de plusieurs niveaux de résolution.

Le problème qui reste est de savoir comment arriver à visualiser la totalité de cette image à partir du Web ?

La réponse à cette question nous a conduit à la mise en place de l'interface Web en utilisant les scripts CGI.

## **INTERFACE AVEC LE WEB**

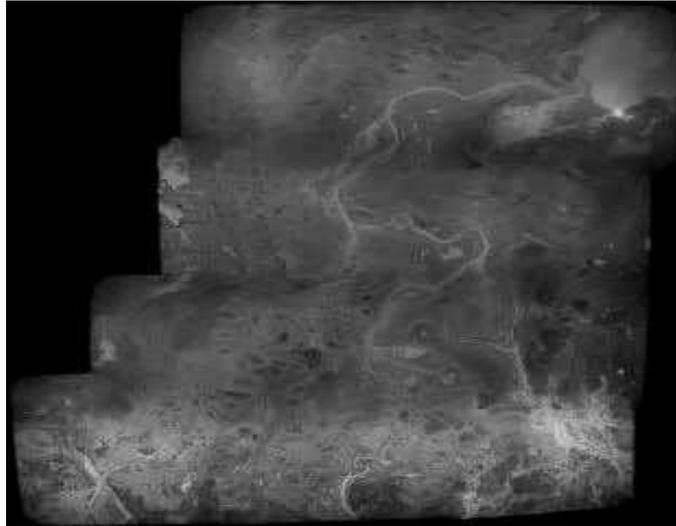
Les mosaïques A, B et C correspondent à trois régions distinctes du Delta Central du Niger. Ces trois régions ont été choisies en fonction de leur intérêt pour l'observation de l'activité de pêche dans le Delta.

Les régions de Walado (mosaïque A) et de Korientze (mosaïque C) ont été choisies pour la complexité de leurs milieux, aux sens naturel (biophysique) et social. Elles ont été sélectionnées en 1995 pour servir de zones test à l'expérience de gestion décentralisée de la pêche, initiée par le gouvernement malien et la FAO avec l'aide de chercheurs de l'IRD. En particulier, ces deux zones, très productives en poissons, étaient l'objet d'une exploitation de pêche très importante, à la fois par les pêcheurs locaux - non sans conflits graves entre communautés voisines - et par des pêcheurs venus du sud du Delta, migrants saisonniers. Ces deux zones sont, elles aussi, particulièrement étudiées dans l'Observatoire de la Pêche. La région de Walado est particulière puisqu'elle connaît deux périodes de crue dans l'année, grâce à une organisation naturelle unique dans la région.

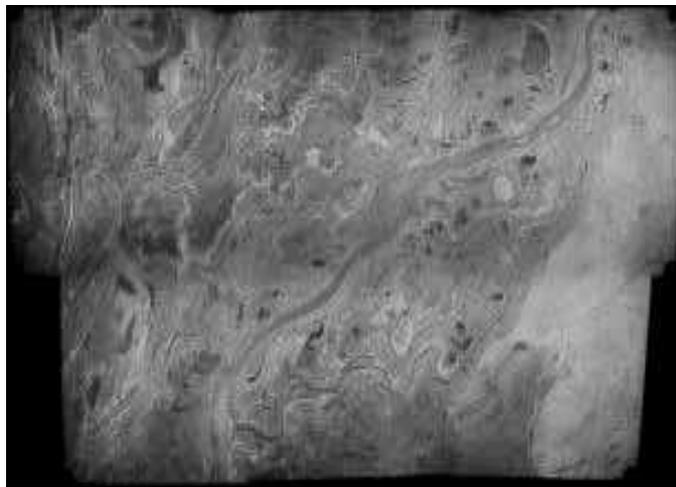
La région de Batamani (mosaïque B) a été choisie en 1988 par les biologistes du programme scientifique sur la pêche dans le Delta (IRD français et IER malien) afin d'étudier le comportement des poissons dans un système " mare-chenal-fleuve Niger" simple et dans lequel il était possible d'effectuer des mesures. D'abord limitée au système hydro-biologique de la mare de Débaré, les études se sont ensuite étendues tout autour dans le cadre des travaux de l'observatoire de la pêche dans le Delta. Une station météorologique simple et plusieurs points de mesures hydrologiques sont maintenus en permanence.

Les travaux qui se poursuivent actuellement sur ces trois zones ont fait l'objet de publications et de rapports dont le détail peut être trouvé sur les sites Internet de l'IRD d'Orléans. Voici quelques références : Quensiere J. 1994 : La pêche dans le Delta Central du Niger. Paris, IRD. Observatoire de la pêche dans le Delta Central du Niger, Morand P. et al. Pamanta O. mémoire de DEA sur la région de Walado, ISFRA, Bamako. Un mémoire de DEA sur la zone de Batamani, Université d'Orléans. Poncet Y. et al. Rapports sur l'expérience de gestion décentralisée de la pêche, Bamako. Poncet Y., Quensiere J., Kintz D. ; plusieurs articles dans des collectifs IRD. Les travaux détaillés qui ont été effectués sur ces trois régions du Delta ont tout particulièrement été exploités pour décrire les mécanismes naturels et sociaux qui sont restitués dans la Modélisation Intégrée du Delta Central du Niger, IRD, Orléans (Kuper, Mullon et Poncet).

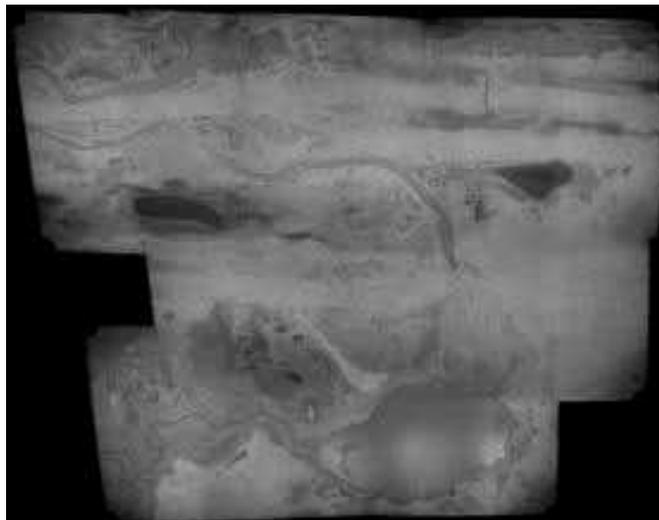
**MOSAIQUE A**



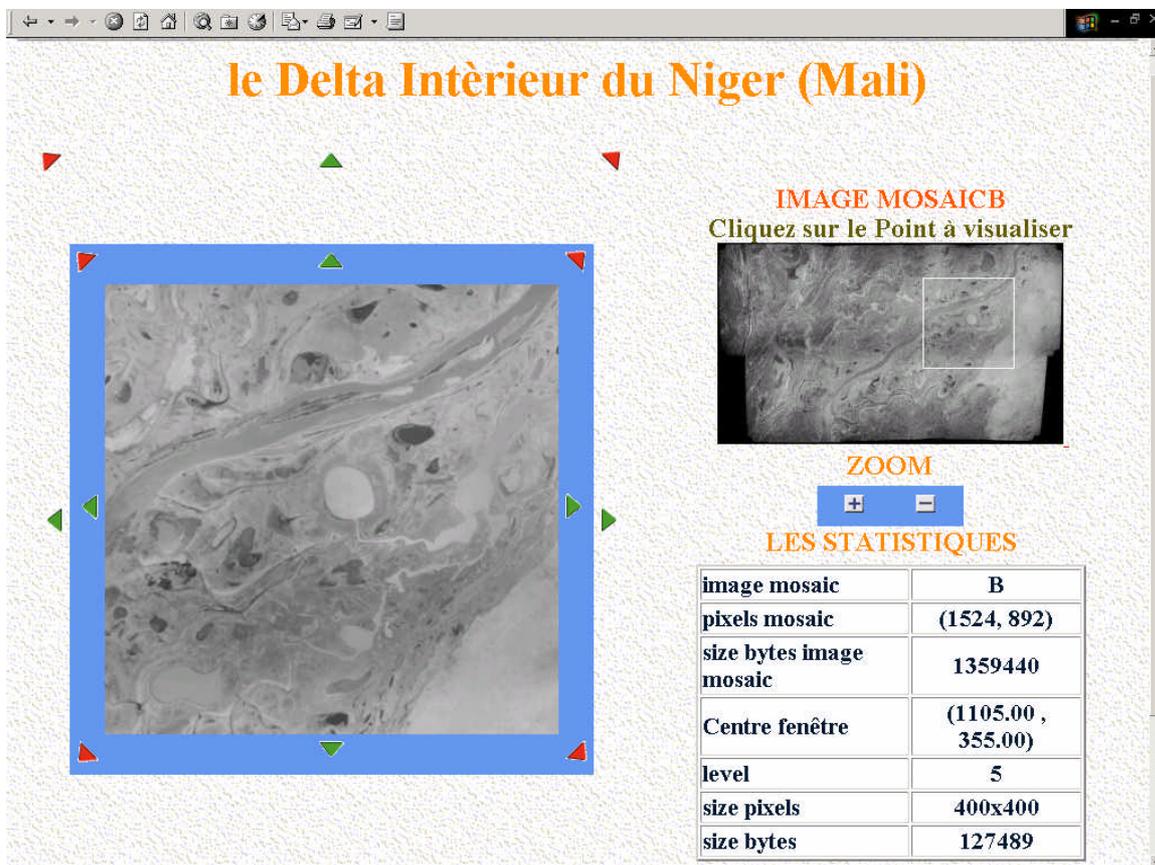
**MOSAIQUE B**



**MOSAIQUE C**



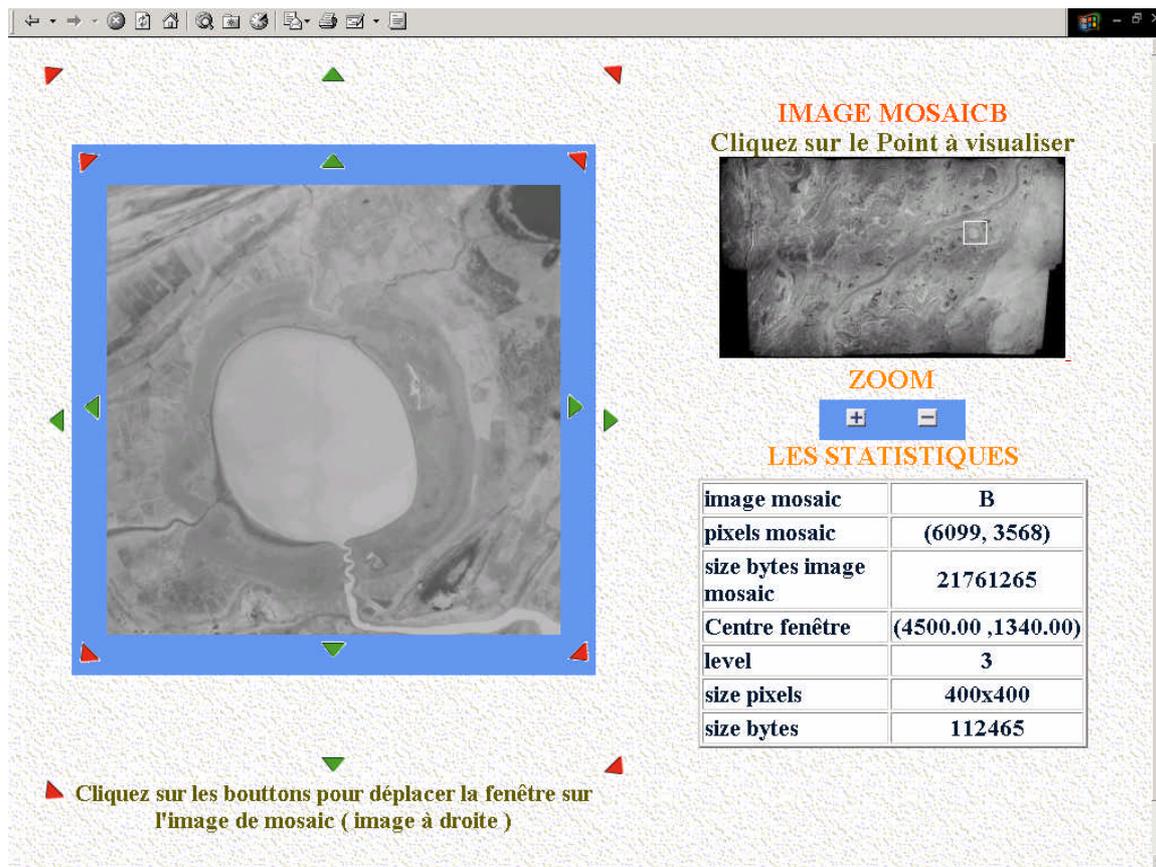
Cette application nous permet de visualiser la totalité de l'image mosaïque (avec toutes les résolutions disponibles). Cette image est stockée sur un serveur Web. La page d'accueil présente les régions concernées ainsi qu'un commentaire leur intérêt géographique. En sélectionnant une région, on pointe vers la page ci-après.



Nous avons à gauche l'image mosaïque correspondante à la région que l'on veut observer. L'image plus petite en haut à gauche permet de sélectionner une zone particulière. Les différentes flèches offrent des possibilités de navigation sur l'image de haute résolution. A droite en bas sont affichées des statistiques concernant l'image affichée ces statistiques sont mises à jour dynamiquement.

Enfin une fonction de ZOOM permet de changer la résolution de l'image.

### Effet du ZOOM niveau 3



### MISE EN LIGNE D'OUTILS DE TRAITEMENT D'IMAGE.

L'objectif est de donner aux utilisateurs distants la possibilité d'accéder par le WEB à des outils et algorithmes de base pour traiter en direct leurs images.

La future version offrira la possibilité à l'utilisateur de poster sa propre image sur le serveur, d'effectuer le traitement et de télécharger le résultat sur son ordinateur personnel.

Nous présentons ci-dessous un extrait du travail en cours.

Choisir le numéro de l'image de test

0 1 2 3 4 5 6 7 8 9 10

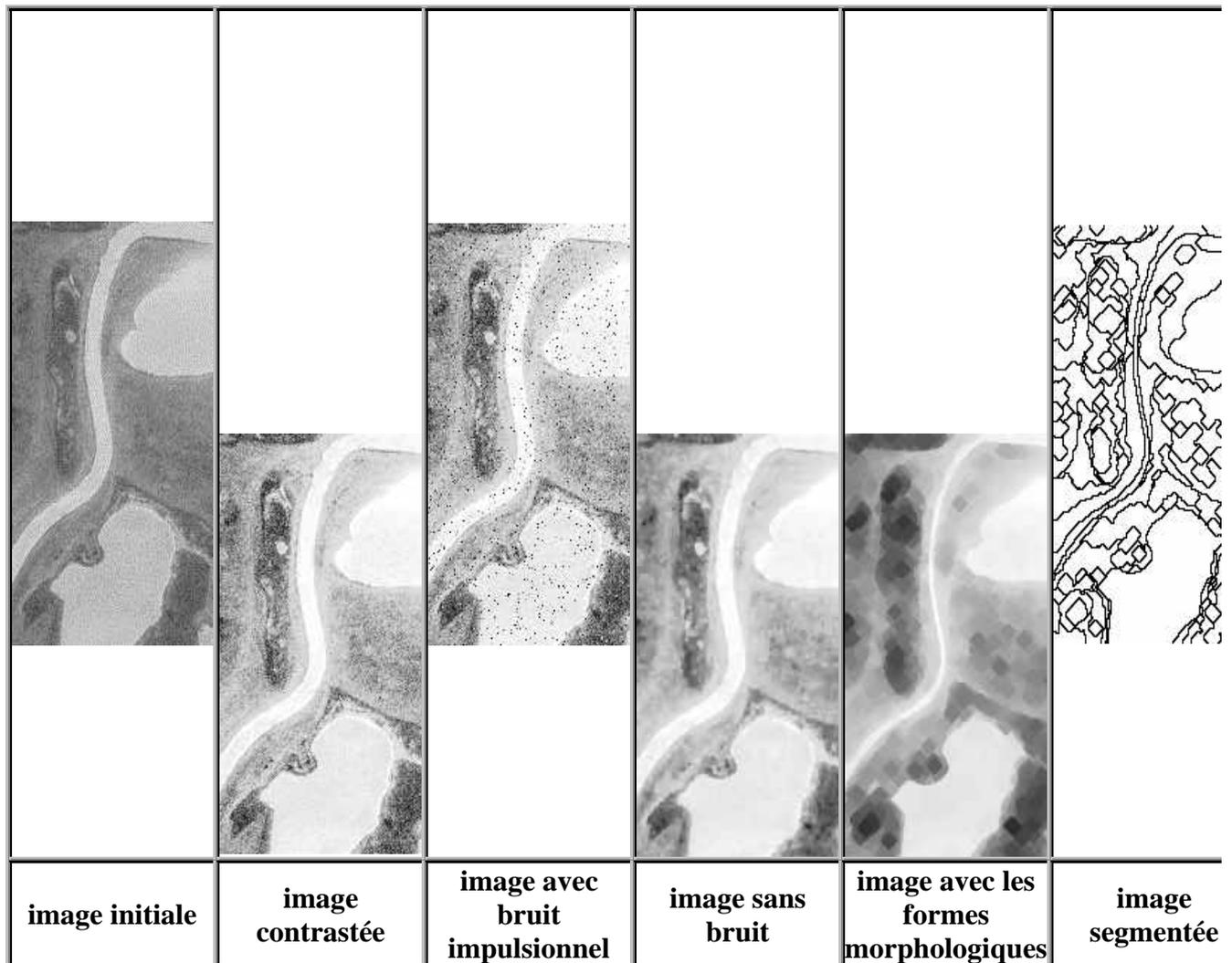
traitement à subir pour une segmentation

Tapez la valeur de Lambda :

On fait subir à une image initiale les traitement suivants

- **Rehaussement de contraste**
- **Ajout de bruit**
- **Suppression de bruit par le filtre médian**
- **Mise en évidence de forme morphologique**
- **Segmentation**

Voici une récapitulation des résultats obtenus avec les différents traitements :



## BIBLIOGRAPHIE

1) Multiple View Geometry in Computer Vision, *R. Hartley & A Zisserman*, Cambridge University Press, June 2000,

2).*Capel, D. and Zisserman, A* : Automated Mosaicing with Super-resolution Zoom

3) *Messmer, B.*: Efficient Graph Matching Algorithms for Preprocessed Model Graphs, 1996

4) *Messmer, B., Bunke, H.* Fast error-correcting graph isomorphism based on model precompilation

5) *B.T. Messmer and H. Bunke.* Subgraph Isomorphism in Polynomial Time

6) Vision par Ordinateur :Outils fondamentaux *Radu HORAUD et Olivier MONGA*