Guénon Marie et Favreau Jean-Dominique

VIM / Master SSTIM Polytech’ Nice Sophia Antipolis

15/01/2013

Calcul de Mosaïques

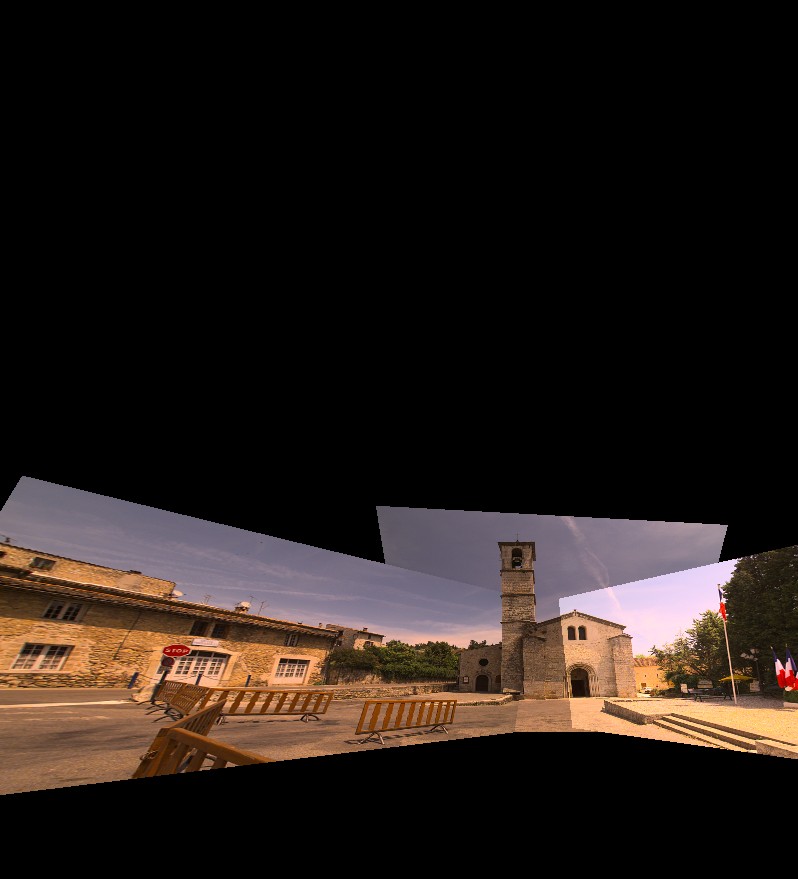


Table des matières

[Introduction 2](#_Toc376622814)

[Calculs théorique de l’homographie 3](#_Toc376622815)

[Passage à la pratique, avec un fichier 4](#_Toc376622816)

[1. Lecture du fichier d’appariements 4](#_Toc376622817)

[2. Calcul de la matrice d’homographie 6](#_Toc376622818)

[3. Recollement des images 7](#_Toc376622819)

[Passage à la pratique, sans fichier 9](#_Toc376622820)

# Introduction

Le but est de calculer la déformation H qu’il a été nécessaire d’appliquer à une image pour que les points d’appariement qui ont été fournis coïncident sur les deux images. C’est cette matrice d’homographie H qui nous permettra de recoller les deux images ensemble grâce à leurs points d’appariement.

**appariements**

H



# Calculs théorique de l’homographie

Considérons deux points et sur deux images différentes. On suppose que la deuxième image a subit une déformation H que nous cherchons à déterminer.

Nous posons :

, et

ayant subit une déformation H pour recoller à , nous avons :

Ce qui est équivalent à :

Or :

Où est la i-ème composante du vecteur .  
Et on pose

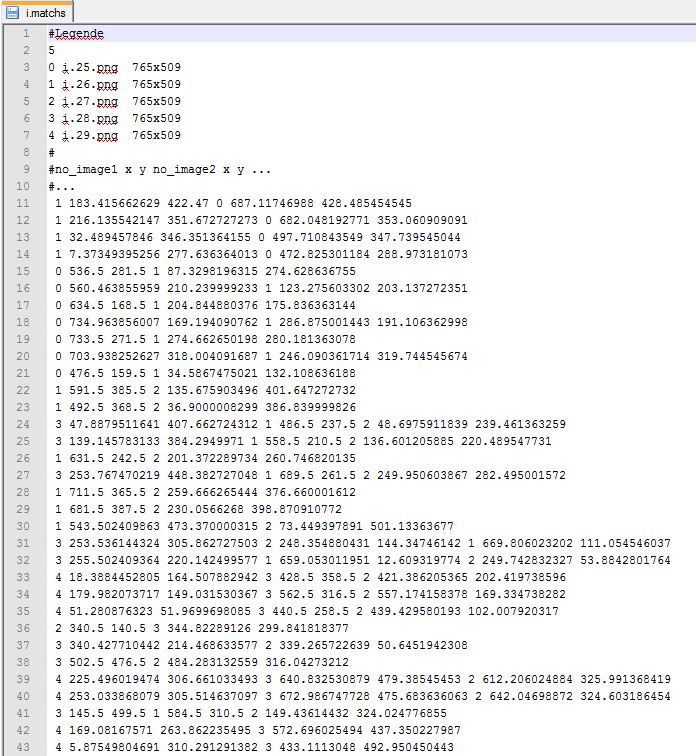
Nous avons donc maintenant à résoudre un problème qui s’écrit : où *X* est l’homographie que l’on cherche sous forme de colonne.

Ce qui n’est possible que si nous avons au moins 3 appariements. Et donc un jeu de neuf équations.

# Passage à la pratique, avec un fichier

## Lecture du fichier d’appariements

La liste des appariements est enregistrée dans un fichier qui a pour extension ".matchs". Ce fichier a une syntaxe bien particulière qui nous permet de lui appliquer un algorithme qui extrait l’intégralité des informations qui nous sont utiles pour le traitement des images que nous aurons à faire par la suite.



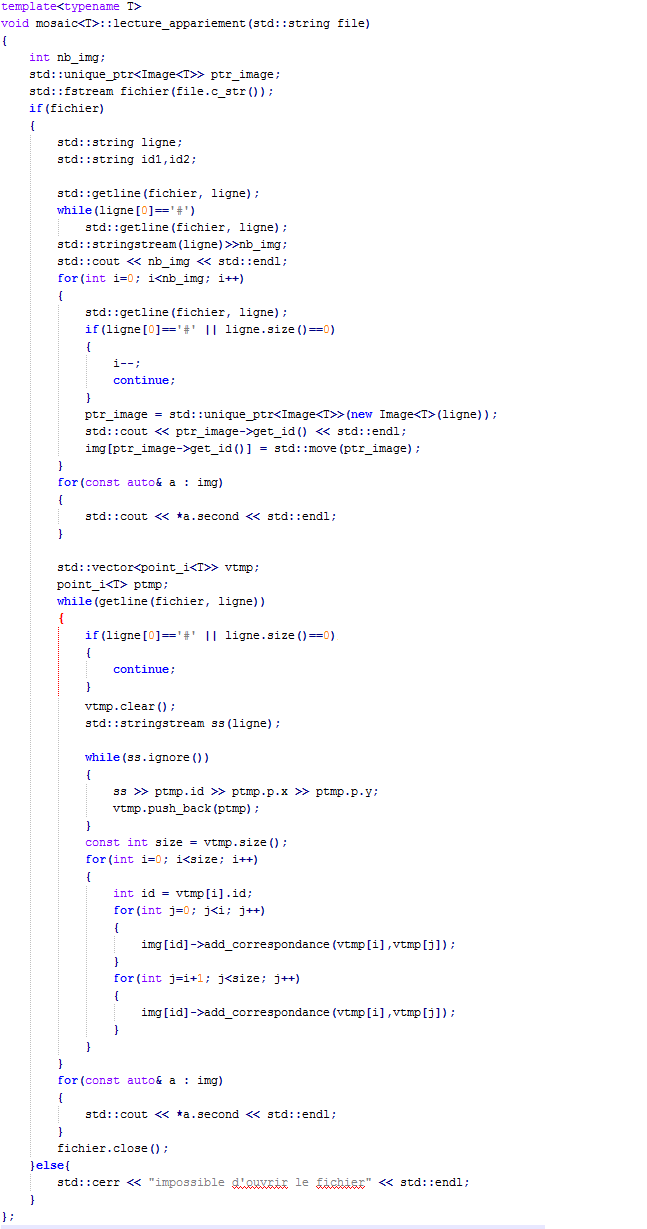
Chaque appariement est écrit de la manière suivante :  
n° de l’image de référence / coordonnée en x / coordonnée en y / n° de l’image à recoller / coordonnée en x dans l’image à recoller / coordonnée en y dans l’image à recoller

Les 3 dernières étapes sont à répéter autant de fois qu’il y a d’images partageant ce point d’appariement.

Liste des appariements

Liste des images à traiter avec la syntaxe suivante :  
n° de l’image / nom de l’image / largeur / hauteur

Nombre d’images à traiter



On associe les correspondances à l’image dont on a récupéré l’id.

On ajoute les correspondances dans la mémoire temporaire

On ignore les lignes qui sont en commentaire ou les lignes vides

On récupère les informations sur les images, que l’on garde en mémoire.

On itère sur les lignes pour récupérer toutes les caractéristiques des images

On récupère le nombre d’images à traiter

On ignore les lignes qui sont en commentaire et on n’incrémente pas le compteur

On ignore les premières lignes qui sont en commentaire

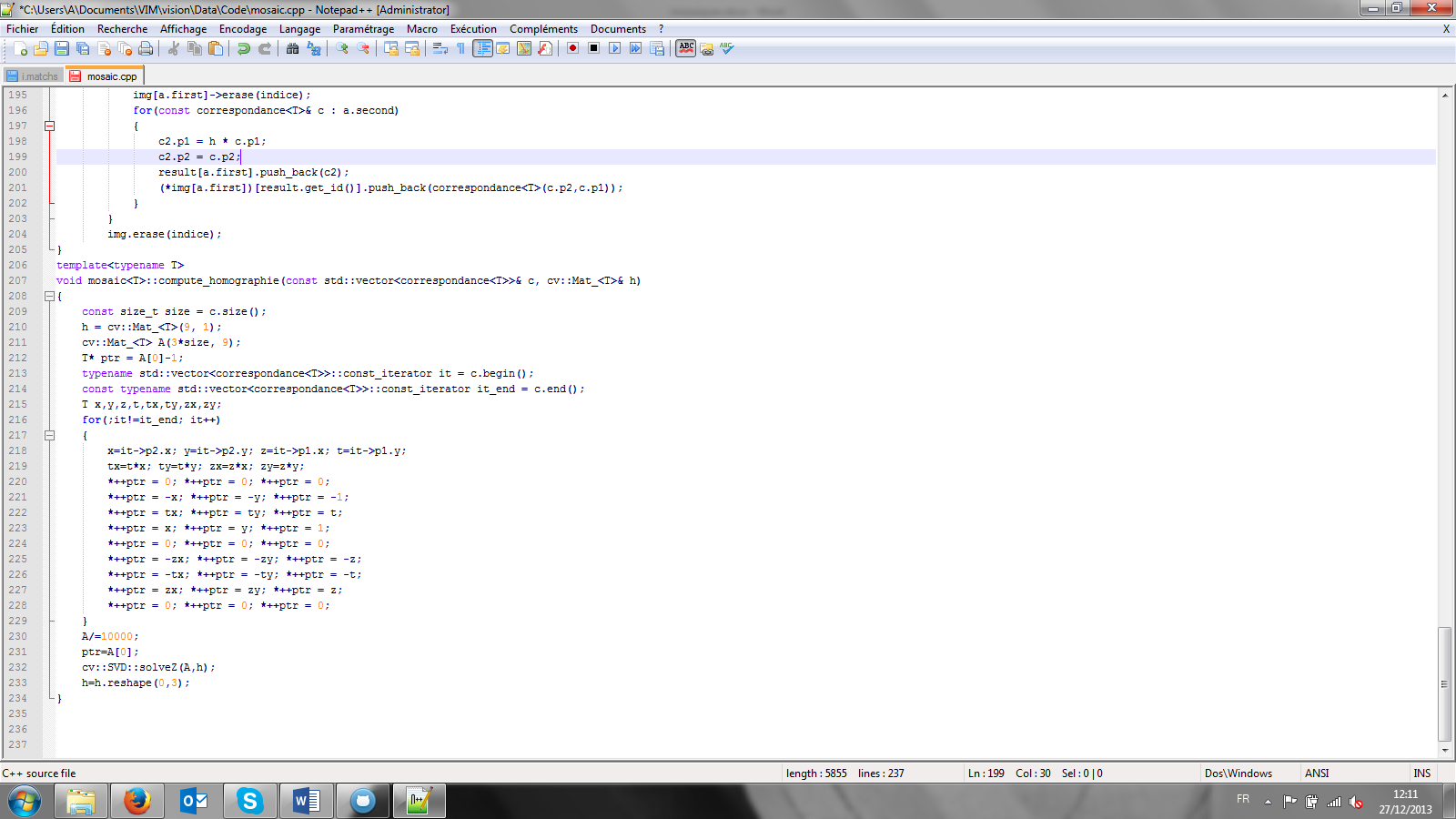
On vérifie que le fichier passé en paramètre existe

## Calcul de la matrice d’homographie

Comme nous l’avons vu précédemment, nous cherchons à résoudre l’équation suivante :

Où .

Ce qui nous donne l’algorithme suivant :



Redimensionnement de H pour avoir une matrice 3x3

Résolution de l’équation

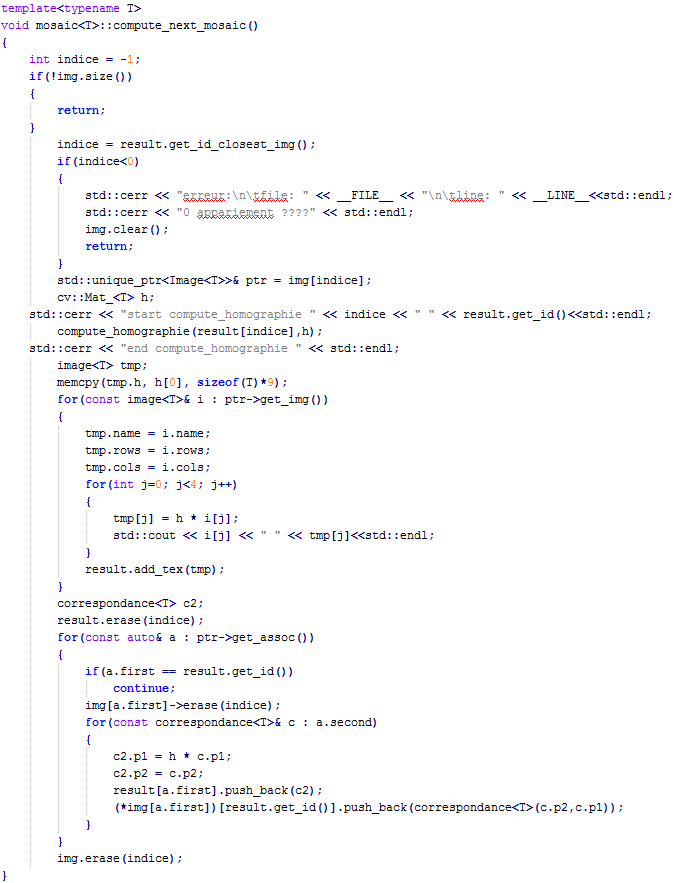
On divise A par 1000 pour éviter d’avoir des éléments d’ordre de grandeur trop différent dans A et améliorer ainsi la stabilité de notre algorithme.

Initialisation des paramètres

On initialise la matrice A

## Recollement des images

Pour effectuer le recollement des différentes images, il s’agit en fait de calculer l’homographie entre l’image de référence et l’image que l’on souhaite ajouter. En règle générale, l’image de référence est l’image dans laquelle sont déjà recollé certaines images. Si nous n’avons pas encore effectué de recollement, nous avons choisi de prendre comme première image de référence celle où il y a le plus de correspondance.



On efface l’image que l’on vient de traiter des futures images à traiter.

On calcule les nouvelles correspondances entre les images restantes et la nouvelle image de référence

On colle l’image la plus proche sur l’image de référence grâce à l’homographie

On calcul l’homographie entre l’image de référence et l’image la plus proche

Si cet indice est nul, ce qui signifie que l’image de référence n’a pas d’appariements avec les images restantes, on envoie un message d’erreur

On récupère l’indice de l’image la plus "proche" de l’image de référence, c’est à l’dire l’image qui a le plus de correspondance avec notre référence

Ce qui nous donne des résultats du type :

|  |  |  |
| --- | --- | --- |
| Image de référence | Image à recoller | résultat |
| C:\Users\A\Documents\VIM\vision\Data\Mosaics\Eglise\i.27.png | C:\Users\A\Documents\VIM\vision\Data\Mosaics\Eglise\i.28.png | C:\Users\A\Documents\VIM\vision\Data\Code\screenshot5.jpg |
|  |  |  |

Nous avons décidé de garder une légère transparence dans les images recollées pour pouvoir voir si le recollement se fait correctement. C’est pour cela qu’en regardant de près les images obtenues, nous pouvons constater l’apparition d’un léger flou aux endroits les plus éloignés des points d’appariement donnés. Ce qui est logique, puisque ce n’est pas à ces endroits que les deux images sont censées se correspondre parfaitement.

# Passage à la pratique, sans fichier

Dans ce cas, nous n’utilisons plus de fichier donné par l’utilisateur et où est listée la liste des appariements entre les différentes images. Pour générer la liste des appariements, nous utilisons une interface qui permet à l’utilisateur de choisir les correspondances entre deux images en cliquant directement sur celles-ci.  
Notre interface se construit comme suit :

* Une première ligne où sont affichées les images en cours de traitement.
  + A gauche l’image de référence
  + A droite l’image à recoller.
* La deuxième ligne est la liste des images disponibles au recollement. Les images en surbrillance sont celles qui sont en cours de traitement.
* Et enfin, nous avons un dernier espace où s’affiche la liste des appariements sélectionnées entre les deux images en cours de traitement.



