

Rectification, MissStereo: manuel d'utilisation

Pascal Monasse, Neus Sabater, Zhongwei Tang

19 août 2011

1 Introduction

Ce document est un (court) manuel utilisateur des logiciels **Rectification** et **MissStereo**, qui calculent une carte de disparité à haute précision à partir d'une paire d'images stéréo. **MissStereo** est la chaîne complète, rajoutant seulement une étape finale de calcul effectif de la carte de disparité tandis que **Rectification** met la paire d'images en géométrie épipolaire rectifiée.

Remarque : Le nom *MissStereo* vient du fait que ce logiciel entre dans le cadre du projet MISS (Mathématiques pour l'Imagerie Stéréoscopique Spatiale) piloté par le CNES et l'ENS de Cachan.

2 Installation

2.1 Avant de commencer

Les dépendances de ce logiciel sont les suivantes :

- **libpng** et **zlib** pour la lecture/écriture des images au format PNG (<http://libpng.sourceforge.net/index.html> et <http://www.zlib.net/>)
- **libtiff** pour la lecture/écriture des images au format TIFF (<http://www.remotesensing.org/libtiff/>)
- **CMake** pour la construction du programme (<http://www.cmake.org/>)
- Compilateur C++ (GNU **g++**)

La plupart des distributions linux proposent des packages pour ces programmes qui sont faciles à installer (si ce n'est déjà fait). Si vous devez les installer, assurez-vous de bien installer la version pour le développeur (package avec une extension **-dev**), de manière à avoir les fichiers header, et pas seulement les bibliothèques.

2.2 Compilation

La compilation s'effectue en trois étapes :

1. Décompresser l'archive du programme.
2. Lancer **cmake** pour générer les **Makefile**.
3. Lancer **make** pour tout compiler.

2.2.1 Décompression de l'archive

Pour décompresser, une possibilité est de lancer la commande

```
$ tar xzf MissStereo.tgz
```

2.2.2 Lancer cmake

Créez un répertoire de votre choix où installer les programmes, par exemple `build`, et placez-vous dans ce répertoire :

```
$ cd MissStereo
$ mkdir build
$ cd build
```

Lancez `cmake` avec comme argument le dossier de base où sont disposés les sources (il y a un fichier `CMakeLists.txt` dans ce dossier) :

```
$ cmake ../src
```

Ceci teste la disponibilité des dépendances et crée les `Makefile` en cas de succès. Pour créer les programmes en mode optimisé, il faut modifier la variable `CMAKE_BUILD_TYPE`, soit avec

```
$ cmake -D CMAKE_BUILD_TYPE:string=Release ../src
```

soit en utilisant l'utilitaire `ccmake` (notez le double c).

2.2.3 Lancer make

Pour compiler, lancez tout simplement

```
$ make
```

Vous pouvez aussi utiliser l'option `'-j2'` pour lancer deux compilations en parallèle (ou un nombre supérieur si vous avez plus de processeurs). Les programmes générés sont alors situés dans le dossier `bin` et les bibliothèques utilisées par ces programmes dans `lib`. Par exemple, vous avez entre autres les fichiers `lib/libNumerics.a` et `bin/rectify`.

Par défaut, les bibliothèques sont statiques. Si vous préférez produire des bibliothèques dynamiques, vous pouvez le faire en mettant à ON la variable `BUILD_SHARED_LIBS`, soit en ajoutant l'option au lancement de `cmake`

```
$ cmake -D BUILD_SHARED_LIBS:BOOL=ON ../src
```

soit en utilisant l'utilitaire `ccmake` (notez le double c).

3 Utilisation

3.1 Mise en place

Le programme est constitué de modules exécutables indépendants qu'il faut lancer dans un certain ordre. Pour cela, un script, `MissStereo.sh`, est fourni. Vous pouvez le lancer d'où vous voulez, ou par exemple le copier dans votre propre répertoire `${HOME}/bin`. Assurez-vous qu'il est bien exécutable :

```
chmod +x MissStereo.sh
```

et son lancement sera directement :

```
MissStereo.sh [arguments]
```

Vous devez informer le script de l'endroit où trouver les programmes exécutables dont il a besoin. Vous pouvez le faire avec la variable d'environnement `MISS_STEREO_PATH`. Pour initialiser cette variable à partir d'un shell Bourne, utilisez

```
export MISS_STEREO_PATH=${HOME}/MissStereo/build/bin
```

Vous devez le faire dans chaque session shell à partir de laquelle vous lancez `MissStereo.sh`. Pour que ce soit automatique, vous pouvez ajouter cette ligne dans votre fichier `.bashrc`.

Une autre possibilité est de mettre le chemin en question directement dans la liste de vos chemins par défaut (variable `PATH`), comme par exemple :

```
export PATH=${HOME}/MissStereo/build/bin:${PATH}
```

Cependant, ce faisant vous "polluez" votre espace de noms d'exécutables. A noter en particulier qu'un des programmes s'appelle `size`, mais qu'il n'a rien à voir avec le `size` de `bin-utils`. Avec un appel comme ci-dessus, vous appelez le programme de `MissStereo` par défaut, pour utiliser celui de `bin-utils` vous devez donner son chemin complet.

Ce programme produit des images et des fichiers intermédiaires qui sont utiles, mais ne sont pas effacés par le script. Pour ne pas mélanger les résultats de plusieurs expériences, il est conseillé de créer un nouveau dossier pour chaque expérience et de se placer dans ce dossier avant de lancer le script.

La procédure est en tout point similaire pour le script `Rectify.sh`.

3.2 Architecture du programme

Le flot de données est illustré Figure 1. Les entrées sont `im1.png` et `im2.png` et les sorties les fichiers `H_im1.png` et `H_im2.png`. Ces mêmes images mais au format TIFF `float`, donc sans quantification, sont produites pour `MissStereo.sh`.

Un étape supplémentaire propre à `Rectify.sh` est détaillée Figure 2. Elle sert à une meilleure visualisation des résultats.

La partie spécifique de calcul de cartes de disparités de `MissStereo.sh` est illustrée Figure 3.

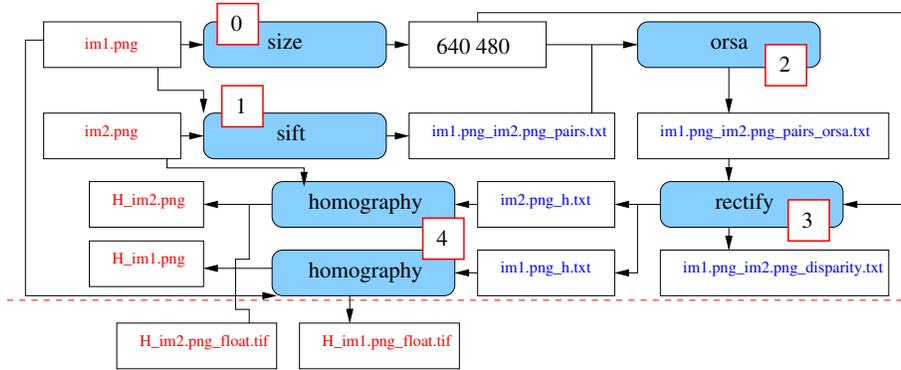


FIGURE 1 – Flot de données de Rectify.sh. La partie sous le pointillé est spécifique à MissStereo.sh.

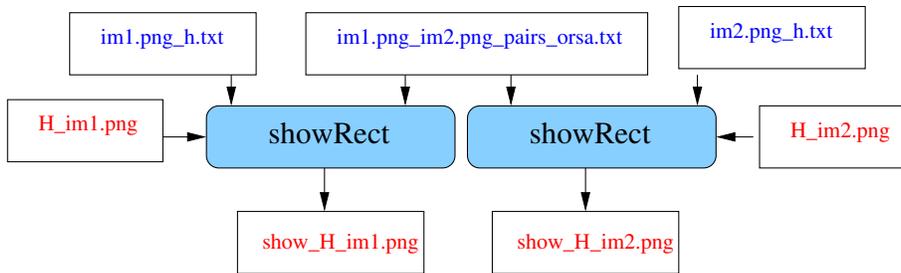


FIGURE 2 – Fin du flot de données de Rectify.sh.

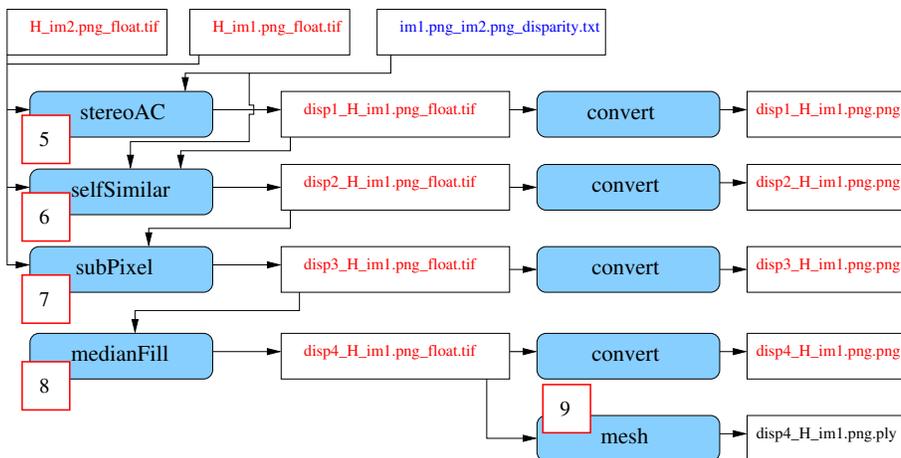


FIGURE 3 – Flot de données de MissStereo.sh.

Les fichiers se terminant par `_float.tif` sont des images TIFF en `float` (attention : peu de programmes savent les lire). Voici la description des différents fichiers :

- `im[12].png` : les images en entrée. Elles doivent être au format PNG, couleur ou niveaux de gris. L'information couleur n'est cependant pas utilisée dans les calculs.
- `im1.png_img2.png_pairs.txt` : fichier texte avec chaque ligne de la forme


```
x1 y1 x2 y2
```

 et indiquant les correspondances de points SIFT trouvées. L'axe y est orienté vers le bas à partir du haut de l'image.
- `im1.png_img2.png_pairs_orsa.txt` : même format que le précédent, mais avec seulement les *inliers* trouvés par l'algorithme ORSA (variante de RANSAC).
- `im[12].png_h.txt` : fichiers texte représentant une matrice 3×3 d'homographie à appliquer aux images pour leur rectification épipolaire.
- `im1.png_img2.png_disparity.txt` : deux nombres entiers qui sont la disparité minimale et maximale des correspondances. Cela permet de restreindre la zone de recherche lors des calculs de corrélation.
- `H_im[12].png` : fichiers des images rectifiées, en 8 bits. Ce sont les images résultats de `Rectify.sh`. Elles ne sont pas utilisées par la suite mais sont simplement plus faciles à visualiser que les suivantes :
- `H_im[12].png_float.tif` : images rectifiées en TIFF `float`. Ce sont celles-ci qui sont utilisées par l'algorithme MARC2 (programme `stereo`).
- `show_H_im[12].png` : mêmes images que `H_im[12].png` mais avec des données superposées. Des lignes horizontales correspondantes sont affichées ainsi que les points SIFT sélectionnés par l'algorithme ORSA.
- `disp1_H_im1.png_float.tif` : carte de disparités en valeur entière et masque.
- `disp2_H_im1.png_float.tif` : idem avec masquage des points auto-similaires.
- `disp3_H_im1.png_float.tif` : carte de disparités sous-pixelliques et masque.
- `disp4_H_im1.png_float.tif` : carte de disparités sous-pixelliques densifiée par filtre médian et masque.
- `disp1_H_im1.png.png` : version en format PNG couleur 8 bits de `disp1_H_im1.png_float.tif` avec les pixels invalides de couleur `cyan`.
- `disp2_H_im1.png.png` : version en format PNG couleur 8 bits de `disp2_H_im1.png_float.tif` avec les pixels invalides de couleur `cyan`.
- `disp3_H_im1.png.png` : version en format PNG couleur 8 bits de `disp3_H_im1.png_float.tif` avec les pixels invalides de couleur `cyan`.
- `disp4_H_im1.png.png` : version en format PNG couleur 8 bits de `disp4_H_im1.png_float.tif` avec les pixels invalides de couleur `cyan`.
- `disp4_H_im1.png.ply` : fichier au format PLY binaire représentant le nuage de points 3D.

Le masque est codé dans les images elles-mêmes : les pixels en dehors du masque de fiabilité prennent la valeur "Not a Number" (NaN), un code autorisé par la norme IEEE 754 concernant le codage des réels à virgule flottante. Dans un code C ou C++, la façon de détecter une telle valeur est la suivante :

```

if(val != val) printf("Not a Number");
if(val == val) printf("A number");

```

A noter que le deuxième test est vrai aussi pour les valeurs $\pm\infty$, également prévues par la norme IEEE. Cependant, ces valeurs ne sont pas utilisées par `stereo`.

Le fichier de points 3D, au format PLY (Stanford Triangle Format), peut se visualiser avec les logiciels libres MeshLab (<http://meshlab.sourceforge.net/>) ou ParaView (<http://paraview.org/>) par exemple. A noter qu'en l'absence de données de calibration, le coefficient de proportionnalité entre la hauteur et l'inverse de la disparité est inconnu. L'axe des z a donc une échelle différente des axes x et y .

3.3 Exemple

```

$ echo $MISS_STEREO_PATH
/home/pascal/MissStereo/build/bin
$ mkdir exp_stereo
$ cd exp_stereo
$ ~/MissStereo/scripts/Rectify.sh ~/MissStereo/data/CarcassonneSmall/im[12].png
sift:: 1st image: 321 keypoints
sift:: 2nd image: 281 keypoints
sift:: matches: 147
seed: 1313740870
Remove 18/147 duplicate matches
Optimized stochastic mode (ORSA).
  nfa=-126.091 size=128 (niter=1)
  nfa=-136.653 size=110 (niter=5)
  nfa=-175.439 size=125 (niter=5)
  nfa=-213.441 size=126 (niter=7)
  nfa=-218.382 size=124 (niter=9)
  nfa=-232.379 size=124 (niter=17)
  nfa=-233.904 size=122 (niter=33)
  nfa=-234.714 size=125 (niter=35)
  nfa=-238.9 size=123 (niter=181)
  nfa=-241.797 size=123 (niter=206)
best matching found: 123 points log(nfa)=-241.797 (500 iterations)
F= [ -4.92494e-09 -2.23627e-07 -2.76683e-05; 3.12134e-07 -8.61556e-08 0.00185006; -5.53
Geometric error threshold: 0.620807
LM iterations: 4 f=902.513
K_left: [ 902.513 0 284.373; 0 902.513 142.5; 0 0 1 ]
K_right: [ 902.513 0 254.529; 0 902.513 142.5; 0 0 1 ]
Initial rectification error: 10.2622 pix
Final rectification error: 0.145428 pix
Disparity: -28 0
$

```

Remarque : les disparités min et max peuvent varier d'une exécution à l'autre du fait du caractère stochastique de l'algorithme ORSA.

4 En cas de problème

Il n'y a pas de hotline dédiée pour ce programme. Néanmoins, veuillez envoyer un email à Pascal Monasse (monasse@imagine.enpc.fr) décrivant votre problème. Si celui-ci se situe lors du lancement du programme sur certaines images, veuillez joindre le fichier de log (par défaut ``${HOME}/.missStereo.log` à moins que vous n'ayez modifié ceci dans le script `MissStereo.sh`). Il est probable que vous ayez aussi à faire parvenir vos images pour analyse du problème.

Liste des fichiers fournis

MissStereo:

```
CMakeLists.txt  doc          mesh.cpp  scripts  userguide.pdf
data            manuel.pdf  orsa.cpp  src
```

MissStereo/data:

CarcassonneSmall

MissStereo/data/CarcassonneSmall:

```
disp4.png  H_im1.png  H_im2.png  im1.png  im2.png
```

MissStereo/doc:

```
manuel.pdf  userguide.pdf
```

MissStereo/scripts:

```
Disparity.sh  MissStereo.sh  Rectify.sh
```

MissStereo/src:

```
CMakeLists.txt  homography  libNumerics  mesh          showRect  subPixel
convert         libIO       libStereo    orsa          sift
dataStereo      libLWImage  libTransform  rectify       size
density         libMatch   medianFill   selfSimilar   stereoAC
```

MissStereo/src/convert:

```
CMakeLists.txt  convert.cpp
```

MissStereo/src/dataStereo:

```
CMakeLists.txt  pca_basis.dat  prolate.dat
```

MissStereo/src/density:

```
CMakeLists.txt  density.cpp
```

MissStereo/src/homography:

```
CMakeLists.txt  homography.cpp
```

MissStereo/src/libIO:

```
CMakeLists.txt  draw.c  draw.h  io_png.c  io_png.h  io_tiff.c  io_tiff.h  nan.h
```

MissStereo/src/libLWImage:

```
CMakeLists.txt  LWImage.cpp  LWImage.h
```

MissStereo/src/libMatch:

```
CMakeLists.txt  match.cpp  match.h
```

MissStereo/src/libNumerics:

```

CMakeLists.txt  homography.h  numerics.cpp  rodrigues.h
computeH.cpp    matrix.cpp    numerics.h   vector.cpp
homography.cpp  matrix.h     rodrigues.cpp

MissStereo/src/libStereo:
CMakeLists.txt  patch.cpp  patch.h

MissStereo/src/libTransform:
CMakeLists.txt  map_image.cpp  spline.h
gauss_convol.cpp  map_image.h   TransformSize.cpp
gauss_convol.h   spline.cpp    TransformSize.h

MissStereo/src/medianFill:
CMakeLists.txt  main.cpp  median_disparity.cpp  median_disparity.h

MissStereo/src/mesh:
CMakeLists.txt  mesh.cpp

MissStereo/src/orsa:
CMakeLists.txt  main.cpp  orsa.cpp  orsa.h

MissStereo/src/rectify:
CMakeLists.txt  rectify.cpp

MissStereo/src/selfSimilar:
CMakeLists.txt  main.cpp  selfSimilar.cpp  selfSimilar.h

MissStereo/src/showRect:
CMakeLists.txt  showRect.cpp

MissStereo/src/sift:
CMakeLists.txt  demo_sift.cpp  filter.cpp  library.h  splines.cpp
demo_lib_sift.cpp  domain.cpp    filter.h    numerics.cpp  splines.h
demo_lib_sift.h   domain.h     library.cpp  numerics.h

MissStereo/src/size:
CMakeLists.txt  size.cpp

MissStereo/src/stereoAC:
CMakeLists.txt  main.cpp  stereoAC.cpp  stereoAC.h

MissStereo/src/subPixel:
CMakeLists.txt  fft.c  fft.h  main.cpp  subpixel.cpp  subpixel.h

```

Liste des auteurs

- Toni Buades (toni.buades@uib.es) : **sift**
- Julie Digne (jdigne@gmail.com) : **mesh**
- Nicolas Limare (nicolas.limare@cmla.ens-cachan.fr) : **libIO**
- Lionel Moisan (Lionel.Moisan@parisdescartes.fr) : **libTransform, orsa**
- Pascal Monasse (monasse@imagine.enpc.fr) : **convert, homography, libIO, libLWImage, libMatch, libNumerics, libTransform, mesh, orsa, rectify, showRect, sift, size, stereo, MissStereo.sh, packaging, documentation**
- Neus Sabater (neussabater@gmail.com) : **stereo**
- Zhongwei Tang (tangfrch@gmail.com) : **libTransform, orsa, sift**