Rectification, MissStereo: user's guide

Pascal Monasse, Neus Sabater, Zhongwei Tang

August 22, 2011

1 Introduction

This document is a short user's guide to software programs `Rectification` and `MissStereo` that compute a disparity map with high accuracy from a stereo image pair. `MissStereo` is the complete pipeline, adding only a final step of disparity computation while `Rectification` puts the image pair in rectified epipolar geometry.

Remark: The name *MissStereo* comes from the origin of the software in the framework of the workshop MISS (Mathématiques pour l'Imagerie Stéréoscopique Spatiale) codirected by CNES (the French national space agency) and Ecole Normale Supérieure of Cachan.

2 Installation

2.1 Requirements

Dependencies of the software are the following:

- `libpng` and `zlib` for reading/writing images in PNG format (`http://libpng.sourceforge.net/index.html` and `http://www.zlib.net/`)
- `libtiff` for reading/writing images in TIFF format (`http://www.remotesensing.org/libtiff/`)
- `CMake` for building the software (`http://www.cmake.org/`)
- C++ compiler (GNU `g++`)

Most linux distributions propose easy to install packages for these (if not already installed by default). If you need to install them, be sure to use the developper's versions (package with extension `-dev`), so that you get header files and not only the libraries.

2.2 Build

The build process has three steps:

1. Decompress the archive.
2. Launch `cmake` to generate `Makefile`.
3. Launch `make` pour compilation and link.

2.2.1 Decompress the archive

To decompress, you can input in a shell the command

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

2.2.2 Launch cmake

Create a folder of your choice where to install the software, for example `build`, and go to that folder:

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

Launch `cmake` with argument the base folder containing the source codes (there is a file `CMakeLists.txt` in that folder):

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

This checks the availability of the dependencies and outputs `Makefile` in case of success. To build optimized version of programs, the variable `CMAKE_BUILD_TYPE` must be modified using

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

or with utility `ccmake` (notice the double `c`).

2.2.3 Launch make

To build, simply type

```
$ make
```

You can also use the option '-j2' to launch two parallel compilations (or more if you have additional cores). The executable files are then in folder `bin` and libraries in `lib`. For example, you get `lib/libNumerics.a` and `bin/rectify`.

By default, static libraries are produced. If you prefer dynamic ones, you can set to ON the variable `BUILD_SHARED_LIBS`, either by adding the option when launching `cmake`

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

or by using the utility `ccmake` (notice the double c).

3 Usage

3.1 Installation

The software is composed of independent executable modules that need to be launched in a certain order. For this purpose a script, `MissStereo.sh`, is provided. You can launch it from whichever folder, or for example copy it in your folder `${HOME}/bin`. Make sure it has the executable bit set:

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

and you can launch it as follows:

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

You need to inform the script of where to find the executable files it needs. You can do it with the environment variable `MISS_STEREO_PATH`. To initialize it from a Bourne shell, use

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

This command must be executed in each shell where `MissStereo.sh` will be called. To make it automatic, you can add this line to the `.bashrc` file in your home folder.

Another possibility is the inclusion in the variable `PATH`), as follows:

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

However, notice that doing so "pollutes" your executable namespace. In particular, one of the programs is named `size` but it has nothing to do with its homonym from `bin-utils`. With a line as above, you call the program of MissStereo by default, to use the one from `bin-utils` you have to input its full path.

This software produces intermediate images and files that are useful but are not erased by the script upon completion. To avoid mixing results from different experiments, it is advised to create a new folder for each image pair and launch the script from such a folder.

For the script `Rectify.sh` the procedure is quite similar.

3.2 Testing your installation

As a test of your build, you can launch the following script:

```
$ ${HOME}/MissStereo/scripts/Test.sh
```

This launches the pipeline on the images `im1.png` and `im2.png` of folder

`MissStereo/data/CarcassonneSmall`

In case of success, you can visually compare the resulting images `H_im1.png`, `H_im2.png`, and `disp4_H_im1.png` in created folder `TestMissStereo` to images in folder `MissStereo/data/CarcassonneSmall`.

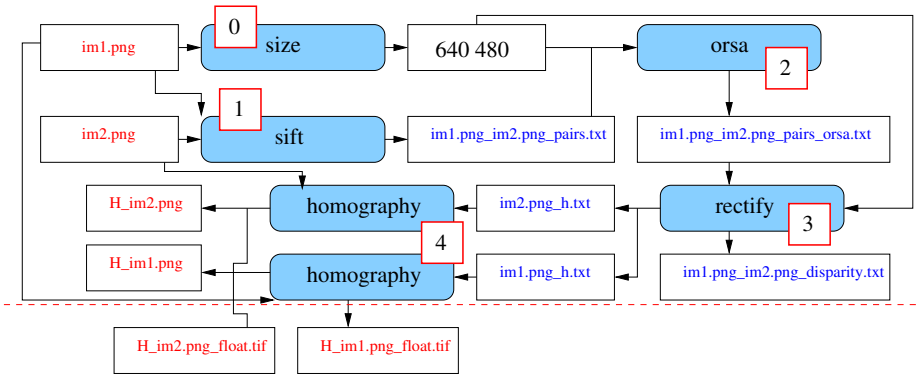

Figure 1: `Rectify.sh` workflow. The part below the dashed line is specific to `MissStereo.sh`.

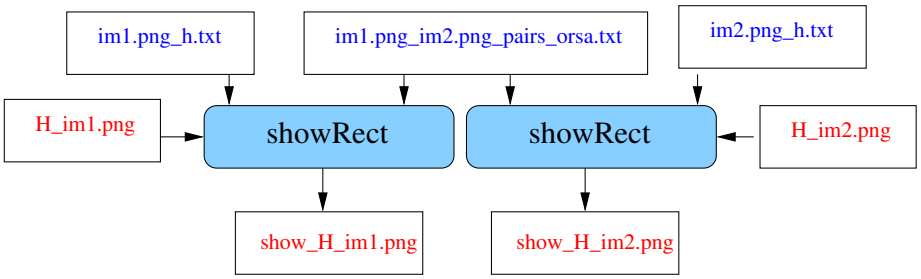

Figure 2: The end of `Rectify.sh` workflow.

3.3 Program workflow

The workflow is illustrated in Figure 1. The input data are `im1.png` and `im2.png`. The same images in TIFF `float` format, thus without quantization, are produced by `MissStereo.sh`.

An additional step specific to `Rectify.sh` is displayed in Figure 2. This is for better visualisation of the results.

The specific part computing disparity maps of `MissStereo.sh` is displayed in Figure 3. This is equivalent to the script `Disparity.sh`.

Files ending in `_float.tif` are TIFF images coded in `float` (warning: few software programs are able to read them). As follows, the description of the different files:

- `im[12].png`: input image pair. They must be in PNG format, color or gray level. However color information is not used during the computation.
- `im1.png_img2.png_pairs.txt`: text file with each line following the format

```
x1 y1 x2 y2
```

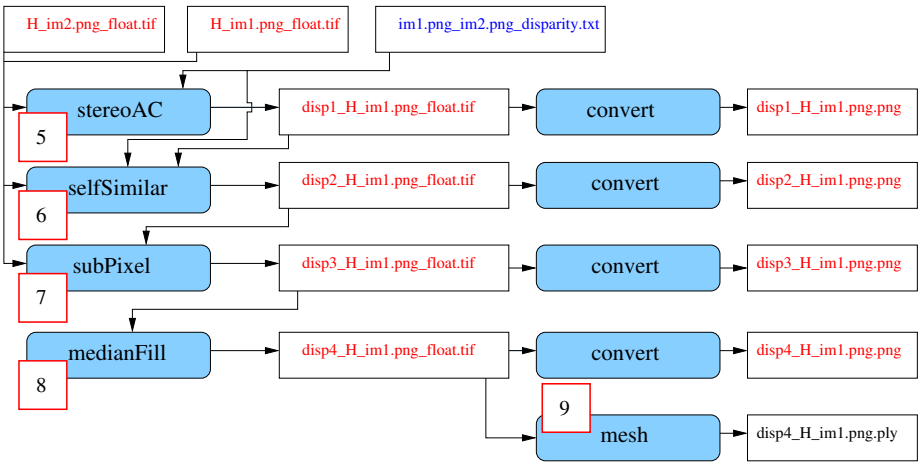

Figure 3: The end of `MissStereo.sh` workflow, corresponding to `Disparity.sh`.

indicating correspondences between found SIFT points. y-axis is oriented downward starting from the top of the image.

- `im1.png_img2.png_pairs_orsa.txt`: same as the previous one but only with inliers found by ORSA algorithm (a RANSAC variant).
- `im1.png_im2.png_disparity.txt`: two whole numbers of minimal and maximal disparity in correspondences. This limits the search area in correlation computation.
- `H_im[12].png`: rectified image files, 8 bits per sample. These are the final images of `Rectify.sh`. They are not used in the following but are easier to visualize than the next ones:
- `H_im[12].png_float.tif`: rectified images in TIFF `float`. These are used in algorithm MARC2 (program `stereo`).
- `show_H_im[12].png`: same images as `H_im[12].png` but with superimposed data visualization. Some corresponding horizontal lines are displayed and SIFT points selected by ORSA algorithm are shown.
- `disp1_H_im1.png_float.tif`: disparity map in integer value and mask.
- `disp2_H_im1.png_float.tif`: the same with self-similar pixels masked.
- `disp3_H_im1.png_float.tif`: subpixel disparity map and mask.
- `disp4_H_im1.png_float.tif`: subpixel disparity map densified by median filter and mask.

- `disp1_H_im1.png.png`: color PNG 8 bit version of `disp1_H_im1.png_float.tif` with cyan color for invalid pixels.
- `disp2_H_im1.png.png`: color PNG 8 bit version of `disp2_H_im1.png_float.tif` with cyan color for invalid pixels.
- `disp3_H_im1.png.png`: color PNG 8 bit version of `disp3_H_im1.png_float.tif` with cyan color for invalid pixels.
- `disp4_H_im1.png.png`: color PNG 8 bit version of `disp4_H_im1.png_float.tif` with cyan color for invalid pixels.
- `disp4_H_im1.png.ply`: PLY binary file represents the 3D points cloud.

The mask is encoded in the images themselves: pixels outside the mask of trust take the value "Not a Number" (NaN), a code authorized by the IEEE 754 norm concerned with floating point real numbers. In a C or C++ code, this can be detected in the following manner:

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

Notice that the second test is also true for values $\pm\infty$, also legal in IEEE norm. However these values are not used by `stereo`.

The 3D points file, with format PLY (Stanford Triangle Format), can be visualized with open source softwares MeshLab (`http://meshlab.sourceforge.net/`) or ParaView (`http://paraview.org/`) for example. Notice that without calibration data, the coefficient of proportionality between height and inverse of disparity is unknown. Therefore, the z axis has a different scale than x and y axes.

3.4 Example

```
$ 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; -5.538
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
$
```

Remark: *min* and *max* disparities can vary between two executions due to the stochastic nature of ORSA.

4 Troubleshooting

Please send an email to the maintainer Pascal Monasse (monasse@imagine.enpc.fr) describing your problem. If it happens when launching the software on certain images, please join the log file (by default `${HOME}/.missStereo.log` unless you have modified this in the script `MissStereo.sh`). It is likely you would also need to send your images for problem analysis.

Usage of different programs

The complete list of executables follows. The last three ones are shell scripts which are actually links to a single file. Invocation name is used to distinguish the requested pipeline. All other executables are built from C++ source files.

- `Usage: size image`
  Return the dimensions in pixels of file `image` (PNG format). This program must not be confused with the standard `/usr/bin/size`.

- `Usage: sift imgIn imgIn2 fileOut [imgOut]`
  Output in text file `fileOut` the matches evaluated by SIFT method between the images `imgIn` and `imgIn2` (PNG format). If `imgOut` is in the argument list, this is an image file where matches will be shown (PNG format).

- `Usage: orsa w h match.txt good_match.txt ntrials verb noseed mode stop`
  Filter out outliers among matches in `match.txt` and output inliers in `good_match.txt`. `w` and `h` must be the common image width and height (the images themselves are not needed). `ntrials` is the number of samples to draw, `verb` is a flag for verbose mode (0 means no display), `seed` the random seed to use (if 0, it is initalized from current time) and `stop` a flag to indicate if we want the algorithm to stop as soon as a meaningful sample is found. The `mode` is an integer with the following meaning:

  0 Try all possible samples (so `ntrials` is ignored). This is only feasible when the number of matches is low.

  1 RANSAC mode with a contrario thresholds. No special sampling strategy.

  2 The same but with new samples drawn among inliers when a meaningful set is found. If `stop` is 0, this is equivalent to mode 1.

  3 Equivalent to mode 2 unless testing all samples results in less than `ntrials`, in which case mode 0 is used.

- `Usage: rectify match.txt w h Hl Hr`
  Take as input a set of good matches `match.txt` and the image dimensions, `w` and `h`, and output the homographies to apply to left (`Hl`) and right (`Hr`) images in the form of 3×3 matrices, stored in Matlab format.

- `Usage: homography image_in H image_out [tiff32_out]`
  Apply homography given by 3×3 matrix in file `H` (Matlab text format) to `image_in` (PNG) and output in `image_out` (PNG format). If `tiff32_out` is given, it outputs the same image but with no quantization as a TIFF float gray-level image. Pixels outside image have value NaN (Not a number).

- `Usage: showRect in.png out.png match.txt left|right H.txt`
  Take as background image `in.png`, superimpose regularly spaced color

dashed lines and small crosses at left or right points in `match.txt` to which the homography given as a matrix in file `H.txt` (Matlab text format) is applied. The ouput image file (PNG format) is `out.png`. The argument before `H.txt` must be the string "`left`" or "`right`".

- `Usage: stereoAC imgIn1 imgIn2 dMin dMax dispMapInc [dispMapMax]`
  From rectified images `imgIn1` and `imgIn2` in PNG or TIFF format and a disparity range between `dMin` and `dMax` (integer values), compute disparity map by *a contrario* block matching. It outputs the results in TIFF float image file `dispMapInc` (integer values, rejected pixels having value NaN). If `dispMapMax` is given, it is a second disparity map computed with a simpler NFA criterion.

- `Usage: selfSimilar imgIn imgIn2 dispmin dispmax dispMapIn dispMapOut [ratio_max]`
  Filter out self-similar blocks in TIFF float disparity map `dispMapIn`, output in `dispMapOut` (so there are more NaN pixels). The rectified images are `imgIn` and `imgIn2` (PNG or TIFF format) and the disparity range given by `dMin` and `dMax` (integer values). `ratio_max`, if given, is the Lowe ratio parameter (positive float value), the default is 1.

- `Usage: subPixel imgIn imgIn2 dispMapIn dispMapOut`
  Refine disparity map `dispMapIn` with subpixel accuracy, output in `dispMapOut` (TIFF float images). Rectified input images are in files `imgIn` and `imgIn2` (PNG or TIFF format).

- `Usage: medianFill imgIn imgOut`
  Fill some NaN pixels in TIFF float image `imgIn` by median filter. Output in `imgOut`.

- `Usage: mesh disp_f32.tif image.png out.ply [K_left K_right]`
  Output in PLY file `out.ply` the colored point cloud with z coordinate computed from input disparity map `disp_f32.tif` (TIFF float image) and color taken from `image.png` (PNG format). Normally, the calibration matrices of the stereo pair are given in `K_left` and `K_right` (Matlab text format). Such matrices are evaluated in Fusiello and Irsara's rectification method. If they are not given, a point cloud is still output but the z range is not accurate.

- `Usage: convert im_float.tif im.png [min max]`
  Convert the TIFF float image `im_float.tif` into PNG 8-bit image `im.png`. If `min` and `max` are given (float values), they are to be mapped to 0 and 255 respectively, with linear interpolation in-between. NaN pixels are mapped to a cyan value in the output image.

- `Usage: density im_float.tif`
  In the TIFF float image `im_float.tif`, count and display to standard output statistics about the number of pixels having a value (different from NaN).

- `Usage: MissStereo.sh image1 image2`
  The full pipeline, rectification and disparity map computation. Output file names are deduced from the names of the input images `image1` and `image2` (PNG format).
- `Usage: Rectify.sh image1 image2`
  Rectification of input images `image1` and `image2` (PNG format).
- `Usage: Disparity.sh image1 image2 dispMin dispMax`
  Compute disparity map from image `image1` to `image2` (PNG or TIFF format). The disparity range is given by integer values `dispMin` and `dispMax`.

List of files

```
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
```

List of authors

- 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`