MicMac Tutorial

How to produce a Digital Surface Model

Using aerial photographs

May – August 2018

Alice GONNAUD

Ecole Nationale des Sciences Géographiques (ENSG)

Elaborated during an internship at Université Libre de Bruxelles (ULB)

Table of contents

Lists of figures and tables

List of figures

List of tables

Useful documentation

Please feel free to read the documentation of MicMac at any time if you need more information:

- Full detailed documentation (pdf file) (mathematical equations, description of tools, tutorials (examples on datasets), algorithms), latest version on the GitHub:
  DocMicMac.pdf in https://github.com/micmacIGN/Documentation

- Online documentation (documentation of the MicMac tools, examples, tutorials):
  https://micmac.ensg.eu/index.php/Accueil

- Help in the terminal (documentation of the parameters of the tools):

  ```
  mm3d MicMac_Tool -help
  ```

- MicMac forum (help from other users and developers, frequent errors):
  http://forum-micmac.forumprod.com/

- MicMac reddit (not as active as the forum) (help on errors):
  https://www.reddit.com/r/MicMac/

- MicMac GitHub (code, documentation (papers, tutorials); you can submit an issue if you want to report a bug):
  https://github.com/micmacIGN

Acknowledgments

Thanks to the CIRB (Brussels) for providing the images used to test and produce this tutorial.

Thanks to the research team of the Université Libre de Bruxelles (ULB, Belgium) for their precious help and advice.

Thanks to the research team of the National Institute of Geographic and Forest Information (IGN, France) for their help and for answering my questions. In particular, thanks to the developers of MicMac who provided me with helpful information.

Introduction

The goal of this tutorial is to explain how to produce a Digital Surface Model (DSM) from aerial photographs using MicMac, along with giving information and tips to understand better the MicMac tools. Thanks to this tutorial, you should be able to adapt the steps of the process chain suggested here to specific datasets: assess the quality of the model, choose better values for parameters, understand and correct errors…

This process chain uses georeferenced aerial photographs: the coordinates of the camera centres are known (typically by GPS measurement). It focusses on a chain that does not require using ground control points (GCPs), but they can be easily added to the process to improve the quality of the products. However, using GCPs requires some manual intervention. The process chain presented in this tutorial is entirely automated.[1]

The steps developed in this tutorial lead to the production of a Digital Surface Model (DSM), a 3D model and/or orthophotos. The MicMac DSM is a georeferenced raster (.tif) whose values are the altitudes of the detail of the pixel, in the coordinate system chosen by the user during the processing (it is the system of the absolute orientation of the model). Please note that the produced DSM shows the elevation of the ground surface or of any object above the ground (buildings, vegetation…) as it is produced using the estimated shapes, sizes and positions of what is visible on the aerial photographs. A Digital Terrain Model (DTM) that only shows the ground surface elevation (even where the ground is covered on the images by buildings or vegetation) cannot be produced using photographs as directly as a DSM, precisely because the ground is covered and therefore modelling it requires interpolation. This tutorial only focusses on how to produce a DSM, meaning a model of the ground surface elevation and all objects on it.

It is also possible to produce the orthophotos and a 3D model (point cloud) using this tutorial.

Please not that MicMac is continuously under development, and has been for years. The tools evolve, get more and more exhaustive and new options are regularly available. This has several impacts on the way MicMac is developed and how it can be used.

Firstly, MicMac is made of few very complex tools, each corresponding to a large photogrammetric process (e.g. Pastis, Apero); of several layers of interfaces to these complex tools that are easier to use and that execute more specific processing steps or embed frequently used sets of parameters (e.g. Tapioca, Tapas, Malt); and of interfaces to these interfaces (e.g. C3DC). This may be confusing to anyone not familiar with the software tools, as it may seem that two distinct tools do the same thing, when in fact one tool is just an interface to the other. The user is still free to use either the complex tool or its simplified interface. In general, the interface is recommended as it is easier to use. Using directly the more complex tools enables the user to have better control over the various parameters. However, a novice may not know how to choose these parameters and may prefer the simplified interface with default values. It may also explains why some tutorials or forum threads recommend different tools for the same process. Using a complex tool rather than its simplified interface is recommended when the default parameters of the simplified interface are not appropriate enough for a specific dataset and configuration, and does not produce satisfying results. Complex tools may also simply be recommended on older threads before the easier interface existed.

Secondly, each tool has some mandatory parameters and many additional options. Thus, MicMac allows a high level of customisation, as the user is given the opportunity to apply the most appropriate settings to produce the best

[1] Because it is a manual step, how to use GCPs is not developed in this tutorial. However, please keep in mind that it is possible to use GCPs and that it is an effective way to improve the model georeferencement. See more in

Optional Steps and Alternatives*, Use Ground Control Points*.

models possible. But it also makes it hard to handle. We recommend reading the list of the options in the documentation[2] about each tool used just in case an option may be useful to the user for their specific dataset or desired product, especially if the quality of the product obtained with default values is not satisfying. It is pointless for us to reproduce the documentation in this tutorial. Only some options will be mentioned, but please keep in mind that more exist.

Thirdly, documenting takes time and MicMac is very complex to document, as it is so customisable. The MicMac documentation is still being written and even though hundreds of pages have been produced, it may seem very poor as soon as you try to adapt a process specifically to your data. This tutorial made the best of the information obtained from available documentation, the MicMac forum for users, tests on datasets and answers directly from developers. Please keep in mind that the information provided in this tutorial is limited by what we managed to obtain from these sources, and that MicMac offers countless possibilities. Therefore, this tutorial is improvable, and the process chain presented here may not be optimal for some specific configurations.

The process chain presented in this tutorial uses directly command lines. Graphical interfaces are now available but are not necessary and not fully documented yet.

Some commands (dense correlation in particular) can be run on a GPU for improved performance.

MicMac offers processing tools, not visualization tools (or they are very basic). You may want to download external software to visualize your data and check the quality of your products. For this tutorial, we used MeshLab[3] to visualize point clouds (to check the quality of the sparse and dense reconstructions) and QGIS[4] to study the DSM and the orthophotos (to assess the quality of the georeferencement in particular, and the elevation of the DSM).

---

[2] Online documentation, `-help` command directly in the terminal or pdf documentation. See *Useful documentation*.
[3] Download MeshLab: http://www.meshlab.net/#download
[4] Download QGIS: https://www.qgis.org/en/site/forusers/download

Quick overview of photogrammetric process and MicMac algorithms

This chapter describes the theoretical aspects of photogrammetry MicMac implements (principle and name of the algorithms), so that the reader can have a better understanding of the photogrammetric process, and has the possibility to look for further information on the strategy of MicMac.

The aim of the photogrammetric process is to model a 3D representation[5] of a scene using information of 2D images. Here, we use aerial photographs.

MicMac uses **Structure from Motion (SfM)** algorithms. SfM estimates three-dimensional structures from two-dimensional images, which capture the environment multiple times from different angles of view. The collection of photographs has to gather photographs of various angles of view, distance from the scene, and they must significantly **overlap** (generally, over 60%) so that a same detail is seen from different angles[6]. This **multiplicity of angles** is what allows the 3D reconstruction. A **photogrammetric acquisition** is an acquisition of photographs that meet the requirements of photogrammetry: multiple angles of view, 60% overlapping, homogeneous lighting, unchanged parameters (focal length, focus)… Using a collection of photographs of a photogrammetric acquisition of a scene, SfM estimates the interior parameters of the camera (distortion, principal point…) that define the angle of view of the camera (useful for reconstruction), the relative positions of the camera (positions when the photographs were taken), and the position of the photographed details of the scene.

The first step is to extract feature points, which are pixels identifiable unambiguously in the images. Descriptors are used to characterize the point (such as gradient of colors around the point). The detected points are then compared between two images[7], using functions of **correlation**[8]. If their descriptors are similar (they correlate), it means the points must refer to the same detail on the ground and the points are coupled. They are **tie points**. Tie points are used to estimate the relative orientation of the images: the position of a same detail on two photographs (two distinct points of view) informs on the position of the camera when the photographs were taken.

The algorithm implemented in MicMac to detect and describe local features in images is the **scale-invariant feature transform** (**SIFT**, by David Lowe, 2004).

For correlation, MicMac uses **scale-space pyramids**. The photographs are under-sampled[9]. Correlation is performed firstly on images of lowest resolution (under-sampling), then resolution is improved (intermediate sampling) gradually until full (or almost full) resolution. It allows to combine fast processing and good reliability.

A **bundle adjustment** with the least squares method estimates simultaneously the optical characteristics of the camera employed to acquire the images, the positions (and orientation) of the camera, and the position of the photographed details of the scene, in an arbitrary working coordinate system at first. It involves the tie points and minimization of their projection (the position of a detail on the ground, its position in each photograph and the position of the camera for each photograph must be coherent).

[5] The products are various: 3D point cloud, Digital Surface Model, orthoimage…
[6] Areas photographed only once cannot be modelled in three dimensions.
[7] All possible pairs can be tested, or only some are picked, according to the situation.
[8] Minimizing an energy function.
[9] This explains why you can find under-sampled images amongst the generated files when you run MicMac commands for correlation.

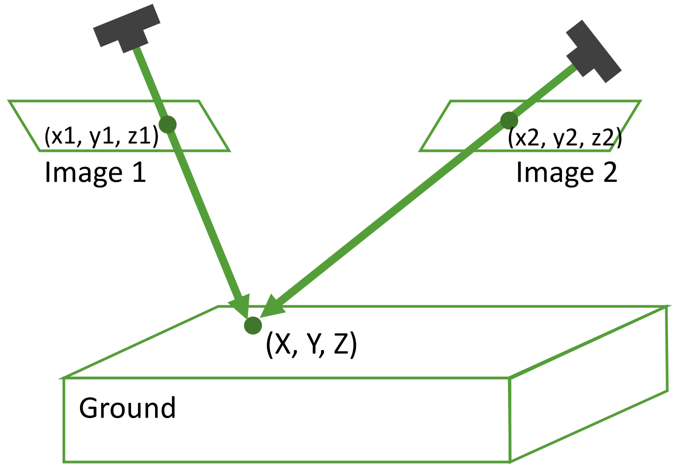


FIGURE 1: THE POSITION OF A SAME DETAIL ON TWO PHOTOGRAPHS INFORMS ON THE POSITION AND ORIENTATION OF THE CAMERA AND THE POSITION OF THE DETAIL ON THE GROUND

Then, the positions can be transformed from the arbitrary system used by MicMac into a geographic coordinate system defined by the user. This transformation is a **3D similarity**. It relies on the knowledge of the position of points in both the initial and the final system (**ground control points, GCPs**). In this tutorial, we use the **position of the camera** for each photograph (typically, from **GPS measurement** during acquisition).

Multi-view stereo image matching (MVSM) is the process of reconstructing a complete 3D model of the scene using the orientation of each image (computed previously) and the position of the details in those images. Corresponding pixels between images provide information on depth. The details of the photographs are projected onto the 3D scene. MVSM is yet again an energy minimization problem. The algorithm measures and finds the solution that minimizes the unlikelihood that two pixels (of two different images) are a unique point on the ground.

How to install MicMac

The MicMac developers recommend using MicMac on Linux, as it was meant to be used on Linux and it generally works better. However, this tutorial was made using MicMac on Windows and no problem due to the OS has occurred.

Please find precise instructions to install MicMac on the online documentation:

https://micmac.ensg.eu/index.php/Install

For this project, MicMac was installed on Windows.

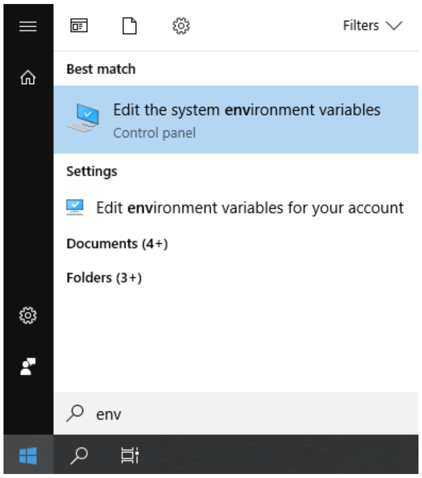
We recommend following the documentation available online:

https://micmac.ensg.eu/index.php/Install_MicMac_Windows .

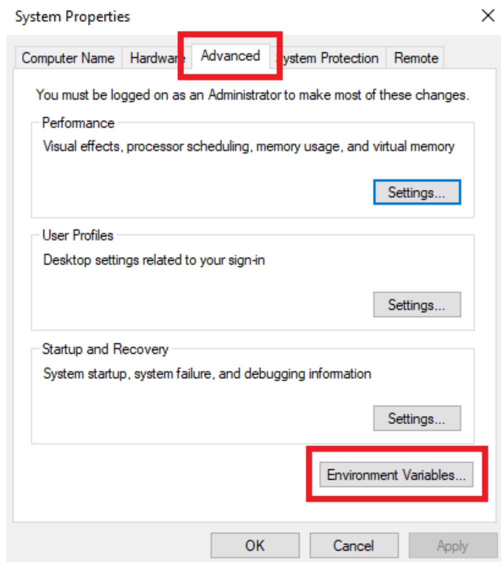
Here are the main steps you can find on the online documentation, with added indications.

Using the binaries (recommended for the newest release):

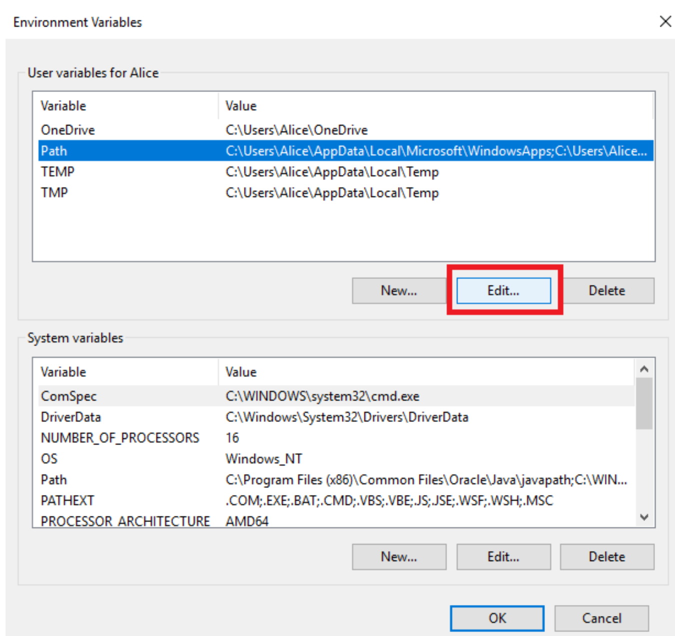
- Download the binaries (for Windows, you want micmac_win.zip):
  https://github.com/micmacIGN/micmac/releases ;
  This tutorial was written using the release *v1.0.beta11* (June 2017).
- Unzip the file where you want it to be. Please avoid any accent or special character in the path.
- Add the path to the system environment variables:
  - Copy the path to the binaries (e.g. *C:\Users\Alice\Documents\MicMac\micmac_win\bin*).
  - Edit the system environment variables:
    You may search directly "environment variables" as the path to the window *Edit the system environment variables* depends on your system.



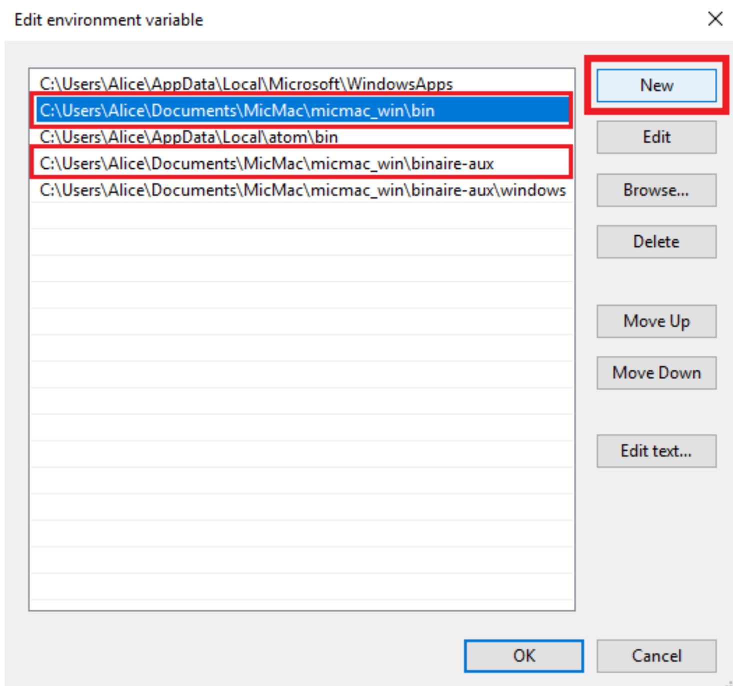
Here is a possible path: *Control panel > System and Security > System > Advanced system settings.*
You will find the window *System Properties*.
Choose *Advanced* > *Environment Variables…*



- Click on the line of the Path to select and then click on *Edit…*



- Click on *New* and paste in the new line the path to the binaries. You also want to add the path to the auxiliary binaries (*binaire-aux* directory).



- Always click on *OK* to close each window.

> **NB: If the following error is raised** at any time when running a program:
> `… not recognized as an internal or external command`
> Please check that the path to the binaries is in the system environment variables.

You want to make sure that the installation is successful by trying the MicMac command in a terminal:

- Type in a terminal the MicMac command:

```
mm3D
```

It should return the list of all the MicMac tools.

- Check that everything installed successfully by checking the dependencies. Type in a terminal:

```
mm3D CheckDependencies
```

Every package (*exiftool, exiv2, convert, proj, cs2cs*…) should be « `found` ».
MicMac uses these external tools to manage image metadata and manage coordinate transformation (projection…), among other things.

Please refer to the online documentation of MicMac if there is any problem when installing:

https://micmac.ensg.eu/index.php/Install_MicMac_Windows

Data and metadata needed to produce the DSM

In order to follow this tutorial, you need the following input data:

- A **dataset of photographs** that meet as closely as possible the **requirements of photogrammetry**: large overlapping (> 60 %), properly exposed (saturation is a loss of information)…
- The approximate **position of the camera** for each photograph (typically, GPS measurement). We will not use ground control points besides these positions.
- You need **at least two photographs to estimate a 3D model** (for stereo view), and **at least three to georeference the model** in an absolute system (required for the 3D similarity). You also need redundant views of the same scene for reliable estimation of the distortion. Generally, you need **at least four or five (overlapping) images for the estimation of the distortion** to succeed. The precise number depends on the configuration: how the images overlap, the texture of the ground… If you do not have enough photos in your dataset, the estimation of the distortion may fail (see more in *Frequent errors*). As a result, if you want to work only on a stereo pair with MicMac, you need to estimate the distortion and the orientation beforehand with a larger set, and apply this first estimation to the pair. If you want to georeference your pair in an absolute coordinate system, you need ground control points in addition to the position of the camera: two positions (for the two photographs) are mathematically not sufficient; you need at least three for the 3D similarity.
  Remember that in general, five overlapping photographs is a safe minimum for satisfying results.
- The **focal length** (in millimetres) and the **dimension of the sensor** (in millimetres) of the camera used for the acquisition, or the focal length in equivalent 35 mm (in millimetres). Please read more about that in *How to add the needed metadata, I) Add the focal length.*

For this tutorial on MicMac, you do not need any other metadata about the acquisition. Parameters such as the principal point or the distortion will be estimated knowing the focal length. However, you imperatively need to know the focal length, as it cannot be determined using other data.

Process chain to produce a DSM using MicMac

Presentation of the process chain

The steps of the process chain to produce a DSM from aerial photographs are the following:

- **Step 1:** Generate files in the MicMac XML format giving the **position of each camera** when the photographs were taken (e.g. GPS measurements). They are produced from the metadata of the photographs or a text file. They will be used to georeference the model.
  The MicMac tool is **OriConvert**.
- **Step 2: Compute tie points.** They will be used to compute relative orientation of the images. Feature points are detected by studying gradients on each image. The feature points are then matched using correlation functions (SIFT algorithm, see *Quick overview of photogrammetric process and MicMac algorithms*).
  The MicMac tool is **Tapioca**.
- **Step 3: Compute inner orientation. Distortion coefficients** and **relative orientation** are computed using least squares adjustment (bundle adjustment).
  The MicMac tool is **Tapas**.
- **Step 4: Compute absolute orientation**. The relative orientation is known in an arbitrary coordinate system (MicMac working system). The images are orientated relatively to one another based on tie points. This model is tilted to fit optimally the absolute coordinates (generated step 1). The relatively orientated model is shifted as a single rigid block, there is no adjustment altering the overall shape of the block. The relative orientation is indeed assumed to be more accurate when computed using tie points rather than using the absolute coordinates. This process allows to combine the precision of relative orientation based on tie points and the (not as accurate) georeferencement.
  The MicMac tool is **CenterBascule**.
- **Step 5: Optional compensation.** If the absolute coordinates are thought to be precisely known, it is possible to adjust the shape of the block obtained using tie points to fit the absolute positions better (the block is only shifted step 4). This step is a compensation of heterogeneous measures: tie points, ground control points (the positions of the cameras here), and distortion coefficients (optional). It should be skipped if the absolute positions (step 1) are not precise, as it gives weight to the absolute positions measured with uncertainty over the coordinates computed more precisely using only tie points.
  The MicMac tool is **Campari**.
- **Step 6: Optional sparse reconstruction**. You can compute the 3D point cloud of tie points to check that the model looks well georeferenced and accurately shaped. This way, you make sure the model is correct before computing the final products (3D point cloud, DSM, orthoimages).
  The MicMac tool is **AperiCloud**.
- **Step 7: Dense reconstruction**. During this step, depth maps are computed and used to produce a 3D point cloud. Depth maps are intermediary products describing the distance between a photographed detail and the position of the camera. The point cloud shows all the photographed details (pixels), which were georeferenced in the absolute coordinate system using the position of the camera when the photographs were taken (computed steps 1 to 5) and the position of the detail in the photographs.
  The MicMac tool is **C3DC**.
- **Step 8:** The depth maps produced step 7 are used to **produce the DSM**. The photographed scene is projected onto a plane, which is the XY plane here as the images are georeferenced (i.e. the horizontal where altitude is null). A tiff raster is produced, whose pixels are the elevation of the detail above the XY plane (Z values). It is also possible to compute the **orthoimages**.
  The MicMac tool is **Pims2MNT**.

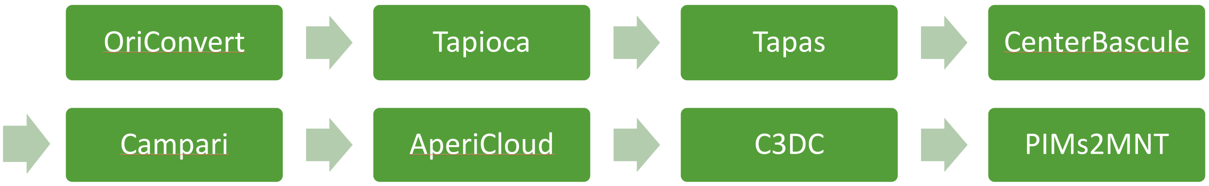


FIGURE 2: PROCESSING CHAIN TO PRODUCE A DSM USING THE TOOLS OF MICMAC

Please note that the camera positions in the absolute coordinate system will be referred to as the "GPS positions", although they may have been acquired by any other means.

Generalities about the MicMac commands

- This tutorial was made using Windows 10: some examples of instructions in this short chapter are specific to Windows.
- You want to **work in the directory that contains the images** to process. Make sure you change directory before running the commands (e.g.):

```
cd C:\Users\Alice\Documents\MicMac\WorkingArea\Dataset1
```

- All the MicMac commands are called using a general "root" MicMac command: **mm3d**.
- You can display documentation with the command **-help** for any tool:

```
mm3d -help
mm3d OriConvert -help
```

  `mm3d -help` displays all the available commands.

- Whenever you run MicMac commands, a log file ***mm3d-LogFile.txt*** that keeps track of all the executed commands is updated (in your working directory).
- When running a MicMac tool such as `Tapas` for inner orientation or `CenterBascule` for absolute orientation, residuals are displayed in the terminal.[10] You may want to save all the information of the terminal into a text file for further analysis or to track errors after the task is completed.
  Just use ">" to specify the output text file after any command line.

```
mm3d Tapas RadialStd ".*tif" Out=Relative > Tapas_shell_residuals.txt
```

- The process chain is automated, meaning that all the instructions can be written in a file to be executed automatically without any manual intervention.
  The chapter *Summary: set of commands ready to be copied and edited* shows a template of such file.
  To write such a file on Windows, just write all the instructions in any text editor and save the file with the ".bat" extension. Just double-click to execute the commands written in the file.

[10] The residuals are automatically saved into an XML file stored in the computed orientation directory. See details further in the chapters describing each step.

- A **set of images** can be defined with a **regular expression**, between quotes.
    - Any character that is not between round or square brackets is common to all the image names
    - (…|…) is the logical "or"
    - […-…] is for a series of values
    - The point "." is for any expression (for unspecified values)
    - The asterisk "*" is to repeat the expression it follows (for unspecified length)
    - Beware that regular expressions are case-sensitive (".*jpg" is different from ".*JPG")

Here are some examples:

- `".*JPG"`: all the JPG images of the folder (every filename ending with "JPG")

    `".*tif"`: all the tif images
- `"IMG(256|325|089).JPG"`: images named IMG256.JPG, IMG325.JPG, IMG089.JPG in the folder

    `"IMG01[2-4]5.JPG"`: images named IMG0125.JPG, IMG0135.JPG, IMG0145.JPG

Regular expressions can be tested with the MicMac command `TestKey`:

```
mm3d TestKey "IMG_.*Y16.tif"
```

Step 0 – Make sure metadata is available before processing

Before proceeding with the actual processing (step 1), you want to make sure you have all the needed data.

a) Focal length

Before starting, you want to make sure either the **focal length** and the dimension of the sensor, or the focal in equivalent 35mm are available: the focal length or the focal in equivalent 35mm must be in the exif metadata of your photographs, and the dimension of the sensor would be in the MicMac XML file *DicoCamera.xml*.

If this metadata is missing or for more details, please refer to the chapter ***How to add the needed metadata, I) Add the focal length.***

b) Position

You will need a **text file with the GPS position** of the camera for each photograph.

To create this file or make sure its format is appropriate to use with MicMac, please refer to the chapter ***How to add the needed metadata, II) Create the position file.***

.

c) Projection

The coordinates of the photographs have to be in a **projected coordinate system**. In particular, if they are GPS data, they cannot remain in the WGS84 system.

Please see why and how to create the **XML file of the projection system** you want to work with in the chapter ***How to add the needed metadata, III) Define a projection system.***

Step 1 – Absolute orientation: using the position of the photographs

This step will:

- Create a MicMac orientation directory in an XML format understandable to MicMac process tools for further process.
  In each orientation file (for each photograph) of this folder, only the positions are known (rotation matrices are null). The positions are the coordinates of the input text file. The rotation matrices will be computed later using tie points (see step 3, Tapas).
- (Optional) Transform the coordinates into the projection system (working in a projection is necessary for MicMac calculations, see ***How to add the needed metadata, III) Define a projection system***).
- (Optional) Generate an XML file storing couples of images that are likely overlapping, given their respective positions. Giving this XML to MicMac to compute tie points will noticeably decrease processing time, especially if the area covered by the photographs is wide and each photograph overlaps with only a small proportion of the other photographs. MicMac will try to match extracted features only in the listed couples of images.

The Micmac command is:

```
mm3d OriConvert File_format Position_text_file Output
(NameCple=Name_xml_file ChSys=initial_system@final_system Options)
```

- **OriConvert** [11] transforms position data from text format to MicMac's XML orientation format.

Mandatory unnamed arguments:

- **File_format** is the format (content of the columns) of the text file that contains the position data of the images.
  See *How to add the needed metadata, II) Create the position file*.
- **Position_text_file** is the text file that contains the position data of the images.
  It must be created as described in chapter *How to add the needed **metadata**, II) Create the position file*.
- **Output** is the name of the output orientation directory (just like every orientation directory in MicMac, it is automatically renamed *Ori-[Output])*.

Optional named arguments:

- **NameCple** computes the XML file **Name_xml_file** storing the couples of images. If the set of images is small and almost all images probably overlap one another, this command is unnecessary, as it will not save time. Otherwise, we recommend you use it.
  You can find a study of this parameter (some processing time values and quality indicators) in the chapter *How to choose parameter values, I) XML file of pairs of images*.
- **ChSys** transforms data from the **initial_system** into the **final_system.**
  Because it is not fully reliable, we recommend you transform your coordinates with an external software instead, and you directly use projected coordinates in the *Position_text_file*.
  See details about this command in *How to add the needed **metadata**, III) Define a projection system*.
- **Options:** other optional named arguments exist, but they are not detailed here. Please refer to the documentation or `-help` command for more information.

[11] OriConvert online documentation: https://micmac.ensg.eu/index.php/OriConvert

Example of the command:

```
mm3D OriConvert "#F=N_X_Y_Z" GpsCoordinates.txt GPS_Georef
NameCple=Couples.xml ChSys=DegreeWGS84@Lambert72.xml
```

Generated files:

- **Directory *Ori-[Output]*** of the orientation files
    - **Files Orientation-[image_name].xml** that contain the position of the camera for each image (one file per image).
      For example:

```
<OrientationConique>
     …
     <Externe>
          Time>317</Time>
          <KnownConv>eConvApero_DistM2C</KnownConv>
          <Centre>152424.029999999999 167513.959999999992 3147.07999999999993</Centre>
          <VitesseFiable>true</VitesseFiable>
          <ParamRotation>
               <CodageAngulaire>0 0 0</CodageAngulaire>
          </ParamRotation>
```

The `<Centre>` tag is the position of the camera. It should be the coordinates of the input *Position_text_file* (or the coordinates after transformation with the option `ChSys`).
The rotation (`<ParamRotation>` tag) is null for now because it will be computed when the relative orientation is estimated (step 3).
You need them for `CenterBascule` (step 4).

- **File *[Name_xml_file].xml*** with the option `NameCple`: file of the pairs of images.
  You need it for `Tapioca File` (step 2).

Step 2 – Computation of tie points

The command to search tie points is `Tapioca`.

If the Couples.xml file has been generated in the previous step, the tie points can be searched only for these couples, which makes the process quicker.

The command line is:

```
mm3d Tapioca Mode Mandatory_Parameters (Options)
```

- **Tapioca** [12] is a simple tool interface for computing tie points. Tapioca is only an interface to `Pastis`, a more complex tool.

There are four main modes:

- **All** tests all possible combinations of images for matching: for each image, tie points will be searched in all the other images. You have to indicate which images you want to process and at which resolution.

```
mm3d Tapioca All PathImages Resolution (Options)
```

[12] Tapioca online documentation: https://micmac.ensg.eu/index.php/Tapioca

- **PathImages** is the regular expression[13] (relative path) referring to the images to process.
- **Resolution** is the reduced width to use for the computation of tie points, in pixels. It is the number of pixels of the longer side of the images. Choosing a lower value will reduce the resolution. Searching tie points in images with lower resolution decreases significantly the processing time. However, it naturally affects the quality of details and feature points may not be detected as effectively.
  You can find a study of this parameter (processing time and quality) in the chapter *How to choose parameter values, II) Resolution for computation of tie points.*
  The user should try a low resolution first, and then try again with a higher resolution if the reconstruction does not seem to work well enough.
  Enter a **resolution of -1 for full resolution**.

For example:

```
mm3d Tapioca All ".*tif" 2000
```

- **MulScale** allows you to choose two resolutions. A first computation of tie points is made for all the pairs of images at very low resolution (faster), then the actual computation, at the high resolution, is done only for the pairs having, at low resolution, a number of tie points exceeding a given threshold. When the resolutions are appropriate, it allows you to combine speed (low resolution) and quality (high resolution) of the computation.

```
mm3d Tapioca MulScale PathImages Low_Resolution High_Resolution
(Options)
```

  - **PathImages** is the regular expression referring to the images to process.
  - **Low_Resolution** is the resolution for the first computation (width of the images in pixels).
  - **High_Resolution** is the resolution for the final computation (width of the images in pixels).
    You can find a study of the resolution parameter (processing time and quality) in the chapter *How to choose parameter values, II) Resolution for computation of tie points.*
  - Optional: **NbMinPt** is the threshold on the number of tie points detected at low resolution (2 by default).

For example:

```
mm3d Tapioca MulScale ".*tif" 2000 6000
```

- **File** is to use a file of couples of images. Only the couples listed in this file will be tested for matching. It saves time, whilst maintaining good quality. It can be computed with `OriConvert` (step 1).
  You can find a study of this parameter (some processing time values and quality indicators) in the chapter *How to choose parameter values, I) XML file of pairs of images*.

```
mm3d Tapioca File XML_file_of_pairs Resolution (Options)
```

  - **XML_file_of_pairs** is the XML file of the couples of images to match, computed with `OriConvert` (step 1).
  - **Resolution** is the reduced width to use for the computation of tie points, in pixels.

[13] See above for more details on how to describe a set of images using a regular expression, chapter *Process chain to produce a DSM using MicMac, Generalities about the MicMac commands.*

It is the same parameter as with the mode `All`.

For example:

```
mm3d Tapioca File Couples.xml 2000
```

- **Line** tests only a certain number of images preceding and succeeding each image[14]. Picking this mode saves time when the acquisition is linear (or circular) such as on facades, but this configuration might be rare for a large aerial dataset.

```
mm3d Tapioca Line PathImages Resolution NumberImages (Options)
```

  - **PathImages** is the regular expression referring to the images to process.
  - **Resolution** is the reduced width to use for the computation of tie points, in pixels.
    It is the same parameter as with the mode `All`.
  - **NumberImages** is the number of adjacent images to look for.
  - Optional: **Circ** is for a circular acquisition: first images will be matched with the last images.

For example:

```
mm3d Tapioca Line ".*tif" 2000 5
```

Generated files:

- **Directory *Homol*** of SIFT descriptors
- **Directory *Pastis*** of tie points
- **Directory *Tmp-MM-Dir*** of temporary files (intermediary results)

Step 3 – Calibration of the camera and relative orientation

This step computes relative orientations and the internal parameter values (principal point of autocollimation, principal point of symmetry, distortion coefficients) to model and then correct distortion. It is the photogrammetric step of bundle adjustment, and it uses the least squares method.

The MicMac command is:

```
mm3d Tapas CalibrationModel PathImages (Out=Name_output_orientation
Options)
```

- **Tapas** [15] computes relative orientation and the internal parameters.

Mandatory unnamed arguments:

- **CalibrationModel** is the calibration model.
  Many models are available. Read about the MicMac calibration models and how to choose one in the chapter *How to choose parameter values, III) Choose a model of calibration.*
  When in doubt, we recommend you use `RadialStd`.

[14] According to the alphabetic order of their names.
[15] Tapas online documentation: https://micmac.ensg.eu/index.php/Tapas

You can also use `AutoCal` to reuse a calibration already computed. See *Reuse a model of calibration* in the same chapter.

- **PathImages** is the regular expression referring to the images to process.

Optional named arguments:

- **Out** allows you to choose the name of the output orientation directory **Name_output_orientation** (automatically renamed *Ori-[Name_output_orientation]).*
- **InCal**, **InOri** are helpful to reuse a calibration or an orientation computed with MicMac. Use them with `AutoCal` as the model of calibration. See *Reuse a model of calibration.*
- Many options allow you to free or freeze some parameters of calibration if needed.
- **SaveAutom** can be set to 'NONE' if you do not want any intermediary results. It means you will not have all the files that could be useful for control in case this step fails, but it saves disk memory.

For example:

```
mm3d Tapas RadialStd ".*tif" Out=Relative
```

While the algorithm is running, the **residuals** of the bundle adjustment are displayed in the terminal. For example:

```
RES:[09533Y16.tif][C] ER2 1.63673 Nn 92.1398 Of 16679 Mul 5861 Mul-NN 5164
Time 0.394
RES:[09534Y16.tif][C] ER2 1.6907 Nn 91.3532 Of 14063 Mul 4492 Mul-NN 3885
Time 0.329
RES:[09535Y16.tif][C] ER2 1.71975 Nn 88.7265 Of 8897 Mul 3034 Mul-NN 2530
Time 0.208
```

- The **name** of the image is between square brackets
- **ER2** is the re-projection error, i.e. the distance (in pixels) between the theoretical position of a corresponding point calculated with the estimated orientation and the position of the point where it is actually detected
- **Nn** is the percentage of tie points kept for the estimation (outliers are cut)
- **Of** is the total number of tie points detected for this image
- **Mul** is the number of multiple tie points, tie points shared by over two images
- **Mul-NN** is the number of multiple tie points kept for the estimation
- **Time** is the time of computation

> **Please check that the residuals are satisfying.**
> You should obtain residuals (ER2) close to 1 px (or smaller), and percentages (Nn) close to 95 % (or higher).

If the residuals are high, it means that the orientation is not precise. Low residuals with low percentage do not mean that the model is precise. If the percentage is low, it means that you have good (low) residuals because all the high residuals are cut.

If the residuals are not satisfying, neither will be the quality of the products. You should try again with other settings (resolution for `Tapioca`, model of distortion for `Tapas`, number of images in the dataset…).

Generated files:

- **Directory *Ori-InterneScan*** for the orientation of analogue photographs (empty for this process chain).

- **Directory *Ori-[Output]*** of the orientation files
  - **File *AutoCal_Foc-[focal_length_in_µm]_Cam-[name_of_the_camera].xml*** which is the file of calibration, with the inner parameters (focal length, principal point, coefficients of distortion…)
  - **Files *Orientation-[image_name].xml*** that contain the position of the camera for each image (one file per image).
    For example:

```
<OrientationConique>
   …
   <Externe>
      <AltiSol>-10.4435311814952883</AltiSol>
      <Profondeur>10.4363418458116026</Profondeur>
      <Time>-1.00000000000000002e+30</Time>
      <KnownConv>eConvApero_DistM2C</KnownConv>
      <Centre>2.0061164767536428 4.95789707706360439 -0.0125983149434508272</Centre>
      <IncCentre>1 1 1</IncCentre>
      <ParamRotation>
         <CodageMatr>
            <L1>-0.999991230842246615 -0.00124477598080110358 0.00399859617446106375</L1>
            <L2>-0.00124098137059239591 0.999998777470747502 0.000951326572901222615</L2>
            <L3>-0.00399977547452811701 0.000946356047207401593 -0.999991553067517303</L3>
         </CodageMatr>
      </ParamRotation>
```

    The <Centre> tag is the position of the camera in the arbitrary working system of MicMac.
    The <ParamRotation> tag is the matrix of rotation of the camera.
    You need them for CenterBascule (step 4).
  - **File *Residus.xml*** with all the residuals for each image and each iteration of the adjustment algorithm (equivalent to the information displayed in the terminal). The final residuals are at the end of the file.
    - <Name> is the name of the image
    - <Residual> is the residual for this image (in pixels), i.e. the distance between the theoretical position of a corresponding point calculated with the estimated orientation and the position of the point where it is actually detected (ideally, should be < 1 px)
    - <PercOk> is the percentage of tie points kept for the estimation (outliers are cut) (ideally, should be > 95 %)
    - <NbPts> is the number of tie points detected for this image (amongst them, only PercOk are kept for the estimation)
    - <NbPtsMul> is the number of tie points shared by over two images

    You should check that the final residuals are satisfying.
- **Directory *Tmp-MM-Dir*** of temporary files (intermediary results) updated

Step 4 – Combine relative and absolute orientation: georeference the model

As explained earlier (see *Presentation of the process chain*), the photographs are oriented relatively using tie points (step 3), and then the whole "block" of relatively oriented images is tilted to meet the absolute positions computed in step 1 (using least squares adjustment). The relative orientation is indeed more accurate when using tie points than when using GPS positions, but the GPS positions are necessary to georeference the model.

In other words, this step transforms the relative coordinates from an arbitrary system (MicMac working system) into the projection system of the final product chosen step 1 (calculation of a 3D similarity).

The MicMac command to shift the model is:

```
mm3d CenterBascule PathImages Relative Absolute Output (Options)
```

- **CenterBascule** [16] transforms a purely relative orientation into an absolute orientation.

Mandatory unnamed arguments:

- **PathImages** is the regular expression referring to the images to process.
- **Relative** is the MicMac relative orientation directory (computed step 3).
- **Absolute** is the MicMac absolute orientation directory (computed step 1) (image centre, i.e. position of the camera when the photograph was taken).
- **Bascule** is the name of the output orientation directory (automatically renamed *Ori-[Bascule])*.

For example:

```
mm3d CenterBascule ".*tif" Relative GPS_Georef Bascule
```

While the algorithm is running, the residuals are displayed in the terminal. For example:

```
Basc-Residual 09533Y16.tif [7.26626,0.769857,-3.63008]
Basc-Residual 09534Y16.tif [8.3172,1.00238,-4.43466]
Basc-Residual 09535Y16.tif [7.22845,-0.131505,-4.22216]
```

They are the offset (in X, Y and Z) in ground unit (e.g. meters) between the estimated position (using relative orientation computed step 3) and the absolute position given initially (positions step 1).

> **Please check that the residuals are satisfying.**
> They should be of the same order of magnitude than the precision of the absolute positions (step 1).
> If the absolute positions are GPS measurements, a few meters are acceptable.

Generated files:

- **Directory *Ori-[Output]*** of the orientation files
  - **File *AutoCal_Foc-[focal_length_in_µm]_Cam-[name_of_the_camera].xml*** which is the file of calibration, with the inner parameters (focal length, principal point, coefficients of distortion…)
  - **Files Orientation-[image_name].xml** that contain the position of the camera for each image (one file per image). In particular, they store the coordinates of the centre in the absolute system (computed step 1) and the rotation (computed step 3) of the camera.
  - **File *Result-Center-Bascule.xml*** of the residuals of the adjustment (equivalent to the information displayed in the terminal).
    - **<Param>** are the parameters of the transformation from the working system of MicMac to the absolute system
    - **<MoyenneDist>, <MoyenneDistAlti>, <MoyenneDistPlani>** are the average distances (total, in Z and in XY) (in ground unit, e.g. meters) between the estimated position (using relative orientation computed step 3) and the given absolute position (positions step 1).
    - **<Worst>** are the worst residuals (in ground unit, e.g. meters) (one image of the dataset)
    - **<Residus>** are the residuals for each image (in ground unit, e.g. meters)

[16] CenterBascule online documentation: https://micmac.ensg.eu/index.php/CenterBascule

Step 5 – Heterogeneous adjustment (Optional)

Step 3 (`Tapas`) has computed the relative orientation, that has been transformed into the absolute system step 4 (`CenterBascule`).

If the absolute coordinates are thought to be precisely known, it is possible to adjust the block obtained using tie points to fit the absolute positions better (the block was only shifted rigidly step 4). This step is a compensation (bundle adjustment) of heterogeneous measures: tie points, ground control points (the positions of the camera here), and distortion coefficients (optional). It should be skipped if the absolute positions (step 1) are not precise, as it gives weight to the absolute positions measured with uncertainty over the coordinates computed more precisely using only tie points.

The MicMac command is:

```
mm3d Campari PathImages Orientation Output(EmGPS=[Ori_GPS,uncertainty]
AllFree=1 Options)
```

- **Campari** [17] is a tool for compensation of heterogeneous measures (tie points and ground control points).

Mandatory unnamed arguments:

- **PathImages** is the regular expression referring to the images to process.
- **Orientation** is the orientation directory you want to perform the adjustment on (absolute orientation computed previously step 4).
- **Output** is the name of the output orientation directory, after compensation (automatically renamed *Ori-[Output]*).

Optional named arguments:

- **EmGPS** specifies that you use GPS positions stored in the **Ori_GPS** orientation directory (step 1) and having the declared **uncertainty** (in ground unit, e.g. meters).
- **AllFree** is a Boolean option (true **=1**, default 0) to re-estimate calibration also (by default, only the tie points and GCPs are re-estimated).
- **GCP** allows you to add ground control points to the estimation.

For example:

```
mm3d Campari ".*tif" Bascule Compense EmGPS=[GPS_Georef,2] AllFree=1
```

Generated files:

- **Directory *Ori-[Output]*** of the orientation files (similar to `Tapas`, step 3)
- **Directory *Tmp-MM-Dir*** of temporary files (intermediary results) updated

Step 6 – Sparse reconstruction (Recommended)

It is now possible to generate the 3D point cloud of the model built so far. It shows the georeferenced tie points and the camera position of each photograph.

[17] Campari online documentation : https://micmac.ensg.eu/index.php/Campari

This step is not necessary to go further and build the DSM, but it is recommended as it allows you to check that the orientation is accurate and that no problem has occurred, before going any further.

The generated file (.ply point cloud) has to be opened with an external software tool (MeshLab [18], CloudCompare[19]…) as MicMac does not allow such visualisation.

The MicMac command to compute the point cloud is:

```
mm3d AperiCloud PathImages Orientation (Options)
```

- AperiCloud [20] is used for generating a visualization of the sparse 3D model and camera positions.

Mandatory unnamed arguments:

- **PathImages** is the set of images to model.
- **Orientation** is the orientation directory of the images to use to compute the point cloud (last orientation directory computed, step 4 or 5).

Optional named arguments:

- **Out** is the name of the output ply file (*AperiCloud_[Orientation].ply* by default).
- **RGB** adds colour (or not) to the points (Boolean, 1 by default).

For example:

```
mm3d AperiCloud ".*tif" Bascule
```

Generated files:

- **File *AperiCloud_[Output].ply*,** the 3D point cloud
- **Directory *Tmp-MM-Dir*** of temporary files (intermediary results) updated

Step 7 – Dense reconstruction: compute the 3D point cloud

The next step is to compute the dense reconstruction. The pixels of the photographs that were not used as tie points are finally added to the 3D model, using their position in the image and the position and orientation of the image defined using the tie points.

It is not necessary to compute the 3D point cloud to obtain the DSM. Please refer to *Alternatives: produce only orthoimages, 3D point cloud or DSM* in *Optional Steps and Alternatives* to read about the other possible commands depending on the products you want. However, even if you are not precisely interested in the 3D model, computing the point cloud is very useful to assess the quality of the reconstruction. It is usually easier to notice problems on the 3D model than on the DSM. Thus, we recommend you produce the point cloud anyway, and this is why this step is presented here. The next step presented to produce the DSM relies on this step (you cannot produce the DSM the way step 8 shows if you do not do step 7). This step (step 7) produces depth maps (raster images that give the distance between the camera and the details of the photographs) to compute the point cloud. These depth maps are reused to produce the DSM (step 8).

---

[18] Download MeshLab: http://www.meshlab.net/#download
[19] Download CloudCompare: http://www.danielgm.net/cc/release/
[20] AperiCloud online documentation: https://micmac.ensg.eu/index.php/AperiCloud

Just as for the sparse reconstruction, **the generated file (.ply point cloud) has to be opened with an external software tool** (MeshLab, CloudCompare…) as MicMac does not allow such visualisation.

The MicMac command for dense reconstruction is:

```
mm3d C3DC Mode PathImages Orientation (OffsetPly=[X,Y,Z] Options)
```

- **C3DC** [21] computes automatically a point cloud from a set of oriented images.

Mandatory unnamed arguments:

- **Mode** is the mode of C3DC, which corresponds to a preset resolution for the point cloud (obviously, the lower the resolution, the faster the processing).
  Read about the existing modes and how to choose one (processing time and quality) in the chapter *How to choose parameter values, IV) C3DC mode.*
  When in doubt, we recommend you use `MicMac`.
- **PathImages** is the regular expression referring to the images to model.
- **Orientation** is the orientation directory of the images to use to compute the point cloud (last orientation directory computed, step 4 or 5).

Optional named arguments:

- **Out** is the name of the output ply file (*C3DC_[Mode].ply* by default).
- **PlyCoul** adds colour (or not) to the points (Boolean, 1 by default).
- **ZoomF** is the final resolution of the images for dense matching. The mode defines its default value. See more in *How to choose parameter values*, *IV) C3DC mode.*
- **OffsetPly** enables you to offset the coordinates of the point cloud, in case the software you use for visualisation does not support large coordinates (many national geographic systems use coordinates with over six figures). **[X,Y,Z]** are numbers to subtract from each coordinate.
- **Masq3D** adds a mask.
- **UseGpu** runs the process on the GPU (CUDA) if your device can, which is faster.
- **FilePair** is to have the pairs of images.

For example:

```
mm3d C3DC MicMac ".*tif" Bascule
```

Generated files:

- **File *C3DC_[Mode].ply***, the 3D point cloud
- New working files in the orientation directory (the directory given as parameter *Orientation* of the command)
- **Directory *PIMs-[Mode]*** of the depth maps and masks for the computation of the 3D point cloud
- **Directory *Pyram*** of the images of the scale-space pyramids useful for the computation
- **Directory *Tmp-MM-Dir*** of temporary files (intermediary results) updated
- **File *MMByPairCAWSI.xml*** of the list of pairs of images used for computation
- **File *MMByPairFiles.xml*** of the list of images used for computation

[21] C3DC online documentation: https://micmac.ensg.eu/index.php/C3DC

Step 8 – Compute the DSM (and the orthoimages)

This step computes the DSM using the depth maps computed previously step 7.

You can also produce the orthophotos.

The MicMac command to compute the DSM is:

```
mm3d Pims2MNT Mode_C3DC (DoOrtho=1 DoMnt=0 Options)
```

- **Pims2MNT** [22] computes a Digital Surface Model (as defined in the introduction of this document: ground and details above ground).

Mandatory unnamed arguments:

- **Mode_C3DC** is the name of the mode used to compute the point cloud (or more specifically, the depth maps) (C3DC, step 7). It is in fact the name of the PIMs [23] directory Pims2MNT uses to compute the DSM.

Optional named arguments:

- **DoOrtho** is a Boolean option to produce also the orthoimages (0 by default).
  You will obtain a collection of images, one orthoimage for each image of your dataset.[24]
- **DoMnt** is a Boolean option to produce the DSM (1 by default). You can choose not to compute the DSM if you only want the orthoimages.

For example:

```
mm3d Pims2MNT MicMac
```

Generated files:

- New working files in the *PIMs-[Mode]* directory (the C3DC mode given as parameter *Mode_C3DC* of the command)
- **Directory *PIMs-TmpBasc*** with intermediary files for computation
  - **File *PIMs-Merged_Prof.tif***, the produced DSM
  - **File *PIMs-Merged_Prof.tfw***, its corresponding file for georeferencement
  - **File *PIMs-ZNUM-Merged.xml***, which describes the produced DSM (scale, origin)
- **Directory *PIMs-TmpMnt*** with intermediary files for computation

With orthoimages (DoOrtho=1):

- **Directory *PIMs-ORTHO*** with the orthoimages and intermediary files for computation
  - **Images *Ort_[image_name].tif***, the orthoimage corresponding to the image *image_name* of the dataset
  - **Files *Ort_[image_name].tfw***, its corresponding file for georeferencement

  If the orthoimage is too heavy, it is tiled. In this case, *Ort_[image_name].tif* links to the tiles *Ort_[image_name]_Tile_[tile_number].tif*. You can merge them together with the tool `ConvertIm`. Please read more about that in ***Optional Steps and Alternatives, Orthomosaic***.

- **Directory *PIMs-TmpMntOrtho*** with intermediary files for computation

---

[22] Pims2MNT online documentation: https://micmac.ensg.eu/index.php/PIMs2MNT
[23] PIMs: Per Image Matchings tool, called by C3DC for correlation
[24] Please note that only areas overlapping with another image can be orthorectified, some areas of the orthoimages may be void.

Optional Steps and Alternatives

Optional tools

Transform DSM into an aesthetic image: GrShade

The computed DSM can be transform into a tif image with artificial shadows for better visualisation of the elevation. **This image is only for visualisation purpose: the value stored in pixels is no longer the elevation.** It mocks a panchromatic photograph with sun projecting shadows, making the elevation stand out.

The MicMac command is:

```
mm3d GrShade DSM (ModeOmbre=Name_shade_mode Mask=Name_mask_file Options)
```

- **GrShade** [25] produces the shading image of depth map.

Mandatory unnamed arguments:

- **DSM** is the path of the image of elevation.
  Use the produced DSM.
  Beware that the DSM is not in the main directory, but in *PIMs-TmpBasc*.

Optional named arguments:

- **ModeOmbre** is the mode of shade that you want to use.
- **Mask** is the mask to use for the output image.
  You can reuse the mask used to compute the DSM.
- **Out** is the name of the output image. By default, it is the name of your input image with "Shade" at the end.

For example:

```
mm3d GrShade PIMs-TmpBasc/PIMs-Merged_Prof.tif ModeOmbre=IgnE Mask=PIMs-TmpBasc/PIMs-Merged_Masq.tif
```

FIGURE 3: EXAMPLE OF IMAGE PRODUCED WITH GRSHADE FROM A DSM

[25] GrShade online documentation: https://micmac.ensg.eu/index.php/GrShade

Generated files:

- The produced image is in the same directory as your input image, under the same name with "Shade" at the end (unless you chose otherwise with the `Out` option).

Transform DSM into an aesthetic image: To8Bits

Also for better visualisation, it is possible to generate instead a tif image with a gradient of artificial colours with the command:

```
mm3d To8Bits Image (Circ=1 Mask=Name_mask_file Options)
```

- **To8Bits** [26] converts a 16-bit or 32-bit image into an 8-bit image, possibly artificially coloured.

Mandatory unnamed arguments:

- **Image** is the path of the image to process.
  Use the produced DSM.
  Beware that the DSM is not in the main directory, but in *PIMs-TmpBasc*.

Optional named arguments:

- **Circ** is a Boolean option to obtain the hypsometric representation with a colour gradient.
- **Mask** is the mask to use for the output image.
  You can reuse the mask used to compute the DSM.
- **Out** is the name of the output image. By default, it is the name of your input image with "_8Bits" at the end.

For example:

```
mm3d To8Bits PIMs-TmpBasc/PIMs-Merged_Prof.tif Circ=1 Mask=PIMs-
TmpBasc/PIMs-Merged_Masq.tif
```

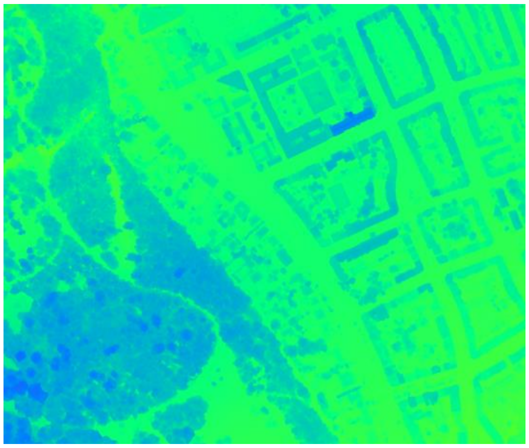


FIGURE 4: EXAMPLE OF IMAGE PRODUCED WITH TO8BITS FROM A DSM

[26] To8Bits online documentation: https://micmac.ensg.eu/index.php/To8Bits

Generated files:

- The produced image is in the same directory as your input image, under the same name with "_8Bits" at the end (unless you chose otherwise with the `Out` option).

> **Tip:** To have **georeferenced images**, you can use the georeferencement file of the DSM (tfw file). All you have to do is to copy and paste it with your image, and name it like your image.

Filter parts of the photographs

You may want to apply a mask to your images, so that MicMac does not spend time reconstructing areas you are not interested in. It might be useful (if not necessary) for challenging locations such as blurred areas, water, vegetation, non-textured areas, details that move from a photograph to another…

You can use the tool `SaisieMasqQT` [27] to outline your area of interest. Please refer to the documentation for more information.

Use Ground Control Points

This tutorial does not develop how to use ground control points, as it is a manual step and the process chain presented here is entirely automated. However, using GCPs is an effective way to improve the georeferencing of the model. If you want to add GCPs to the process, please refer to the MicMac documentation, and in particular see the `SaisieAppuisInitQT` [28] and `SaisieAppuisPredicQT` [29] tools.

You can use `SaisieAppuisInitQT` to define GCPs manually, compute a first approximate orientation, then use `SaisieAppuisPredicQT` to generate automatically the other GCPs based on the first orientation. Correct the automatically generated GCPs when needed and compute the final orientation.

Orthomosaic

You can produce the orthoimages for each image of your dataset (see step 8).

An orthomosaic is one single orthoimage obtained by merging a collection of overlapping orthoimages.

If you want to produce a georeferenced orthomosaic from the multiple (per image) orthoimages, you can use the tool `Tawny`:

```
mm3d Tawny Orthoimages_directory (RadiomEgal=0 Options)
```

- **Tawny** [30] is a simplified tool for generating orthomosaics.

Mandatory unnamed arguments:

- **Orthoimages_directory** is the directory of the orthoimages to combine into the mosaic.

---

[27] SaisieMasqQT online documentation: https://micmac.ensg.eu/index.php/SaisieMasqQT
[28] SaisieAppuisInitQT online documentation: https://micmac.ensg.eu/index.php/SaisieAppuisInitQT
[29] SaisieAppuisPredicQT online documentation: https://micmac.ensg.eu/index.php/SaisieAppuisPredicQT
[30] Tawny online documentation: https://micmac.ensg.eu/index.php/Tawny

If you produced the orthoimages with `Pims2MNT`, they should be in *PIMs-ORTHO*.

Optional named arguments:

- **RadiomEgal** is a Boolean option to perform or not radiometric equalization (1 by default). If your images do not need it, you can save time by setting this option to False.

For example:

```
mm3d Tawny PIMs-ORTHO/ RadiomEgal=0
```

If the orthoimage (per image or global mosaic) is too heavy, it is tiled. In this case, *Ort_[image_name].tif* links to the tiles *Ort_[image_name]_Tile_[tile_number].tif*. You can merge the tiles together with the tool `ConvertIm`:

```
mm3d ConvertIm Image_name (Options)
```

For example:

```
mm3d ConvertIm Ort_09347Y16.tif
```

If the tiles are particularly heavy, the command will not make any difference and the output will remain tiled.

Only the global file *Ort_[image_name].tif* is georeferenced with a tfw file of the same name, tiles are not georeferenced. If you want to georeference your tiles, try and use `ConvertIm`, copy and paste the tfw file of the tiled image before conversion, and rename it to match the output of `ConvertIm`.
If the merging of the tiles fails, you can georeference the tiles manually by writing the appropriate tfw file for each tile, according to the tfw file of the global image and the position and size of the tile in relation to the global image.

Read tie points in text file

With `Tapioca` (step 2), it is possible to use the Boolean option `ExpTxt=1` to have the file of tie points in readable text format instead of binaries. However, this option must be activated too for the other steps depending on this file. For the process chain of this tutorial, the tools `Tapioca` (all modes), `Tapas`, `Campari`, `AperiCloud` and `C3DC` have the option.

You may want to use instead a debug tool `TestLib` that generates a file of all the tie points for each configuration (multiple pairing) of images:

```
mm3d TestLib ConvNewFH PathImages Output (Bin=0 Options)
```

- **PathImages** is the regular expression referring to the images.
- **Output** is the name of the output file.
- **Bin** is the Boolean option for binaries or text format (0 means you want text).

For example:

```
mm3d TestLib ConvNewFH ".*tif" TiePoints Bin=0
```

Generated files:

- **File of the tie points** for each configuration in the directory *Homol*[31] (computed by `Tapioca`), named *PMul[Output].txt*.

Alternatives: produce only orthoimages, 3D point cloud or DSM

You may stumble across other tools than those described in this tutorial (`Malt` or `PIMs` instead of `C3DC` and `Pims2MNT`). The fact that MicMac works in "layers" of interfaces, as explained in the introduction, offers you several possibilities to produce what you want. Here is a glimpse of these other alternatives.

Malt

`Malt`[32] is an interface to the complex tool `MicMac` for image matching. It has three modes:

- **GeomImage**, for a matching in ground image geometry (3D model).
- **Ortho**, for a matching adapted to the computation of orthophotos.
- **UrbanMNE**, for a matching adapted to (urban) DSM.

You can use `Malt GeomImage` after the orientation (`Tapioca`, `Tapas`, `CenterBascule`) to perform dense correlation on a block of images around a master image (all the other images are tied to the master). It is partially manual as you need to choose the master image. Thanks to the master image properly picked, `Malt GeomImage` usually produces better results than the fully automated chain with `C3DC` on a single block of images (typically 5 images).

PIMs

PIMs stands for *Per Image Matchings*.

When you work on an automated process chain and with a large number of images, you want to use the tool `PIMs`[33], which calls `Malt GeomImage` multiple times. With `PIMs` you do not have to pick master images, meaning that it will run automatically for a whole dataset. `PIMs` generates depth maps.

Tools using PIMs

After you perform `PIMs` on your dataset for dense matching (computation of the depth maps for each image), you may want to transform the results of `PIMs` (depth maps) into a point cloud. The tool `PIMs2Ply`[34] does precisely that (or `Nuage2Ply`[35] after `Malt`, which converts a depth map into a point cloud).

`C3DC` (used in this tutorial) calls `PIMs` and automatically computes the point cloud for the whole dataset. It is therefore a simplified interface to MicMac tools for dense correlation and the computation of the 3D model.

[31] "Homol" for homologue points (corresponding points)
[32] Malt online documentation: https://micmac.ensg.eu/index.php/Malt
[33] PIMs online documentation: https://micmac.ensg.eu/index.php/PIMs
[34] PIMs2Ply online documentation: https://micmac.ensg.eu/index.php/PIMs2Ply
[35] Nuage2Ply online documentation: https://micmac.ensg.eu/index.php/Nuage2Ply

If you do not want the point cloud, you can use `Pims2MNT` directly after `PIMs` to compute the DSM. You can tell by the name, it is precisely what it is meant for. In this tutorial, we do call `Pims2MNT` after `PIMs` as `C3DC` calls `PIMs`, although it is not obvious. `Pims2MNT` calls `Malt UrbanMNE`.

With `Pims2MNT`, the Boolean options **DoOrtho** and **DoMnt** allow you to choose whether you want to compute the orthophotos and the DSM.

Summary: set of commands ready to be copied and edited

```
mm3D OriConvert "#F=N_X_Y_Z" GpsCoordinates.txt GPS_Georef
NameCple=Couples.xml ChSys=DegreeWGS84@Lambert72.xml

mm3d Tapioca File Couples.xml 2000

mm3d Tapas RadialStd ".*tif" Out=Relative

mm3d CenterBascule ".*tif" Relative GPS_Georef Bascule

mm3d Campari ".*tif" Bascule Compense EmGPS=[GPS_Georef,2] AllFree=1

mm3d AperiCloud ".*tif" Compense

mm3d C3DC MicMac ".*tif" Compense

mm3d Pims2MNT MicMac
```

Where:

- Step 1: OriConvert computes orientation files with the GPS positions of the images
- Step 2: Tapioca computes tie points
- Step 3: Tapas computes relative orientation and calibration of the camera
- Step 4: CenterBascule transforms the relative orientation into the absolute system
- Step 5: Campari readjusts the absolute orientation
- Step 6: AperiCloud computes the sparse reconstruction
- Step 7: C3DC computes the dense reconstruction (3D point cloud)
- Step 8: Pims2MNT computes the DSM (and the orthophotos)

How to add the needed metadata

I) Add the focal length

The focal length, along with the positions of the camera, is a parameter that has to be known to run the photogrammetric process. It cannot be deduced from the photographs; in fact, it is the other way round. All the other parameters will be deduced from the focal length and the configuration of the photographs (their positions according to what they show).

To apply the collinearity equations and build the 3D model, MicMac needs to know the focal length in pixels. It is determined from the focal length in millimetres and the dimensions of the image known in pixels and in millimetres. Therefore, there are two possibilities:

- If the focal length in millimetres is known, you also need to give the dimensions of the sensor (in millimetres). As the dimensions of the image in pixels is known, the focal length in pixels can be deduced.
- If the 35 mm equivalent focal length is known, the focal length in pixels can be calculated, knowing the dimensions of the images in pixels and that is corresponds to a sensor measuring 36x24 mm[36]. The 35 mm equivalent focal length is the focal length needed to obtain the same angle of view as the one obtained with the focal length and the sensor dimensions of the acquisition, but with a sensor measuring 36x24 mm.

If the sensor dimensions and the 35 mm equivalent focal length are unknown, you may face the following error in MicMac:

```
WARN  !! , for camera Vexcel Imaging GmbH UltraCam Eagle f210 cannot
determine focale equiv-35mm
add it in include/XML_User/DicoCamera.xml
```

As explained in the warning message, you need to add the dimensions of the camera in the XML file ***DicoCamera.xml***. You can find the XML by default in the directory *include/XML_MicMac*. You can modify directly this file, but we recommend you copy and paste it into a new directory *XML_User*, and you modify this copy. When MicMac needs an information from the XML files, it will search in this directory first. In case you update your MicMac version, the content of XML_MicMac will be reloaded and you will lose all the cameras you added. The *XML_User* directory remains unchanged and all your modifications are safe.

Add the camera (new block with a `<CameraEntry>` tag) wherever in the file *DicoCamera.xml*. The dimensions of the sensor (`<SzCaptMm>` tag) are in millimeters. The name (`<Name>` tag) must be exactly the same name as in the exif metadata of the image. It is how MicMac knows which sensor dimensions to use.

The name is the *Camera model* field, for instance:

[36] The name "35 mm" refers to the *35 mm film format* (also known as "full-frame"), a standard dimension of film used for analogue photography. Its height is 35 mm, but the photosensitive area is only 24 mm high because of the perforations along the edges of the film useful to the mechanism to unroll the film. It is 36 mm wide. The use of the 35 mm equivalent focal length is inherited from the analogue era.

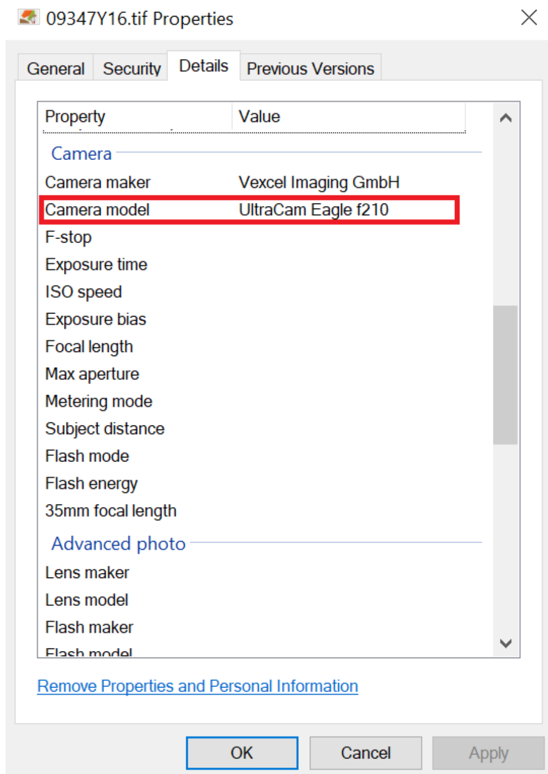


FIGURE 5: NAME OF THE CAMERA IN THE METADATA OF THE IMAGE

If there is no camera model in the metadata, you can add it with the MicMac command `SetExif`[37]:

```
mm3d SetExif PathImages Metadata
```

- **SetExif** [38] is a tool to fill in (or modify) metadata required by MicMac.
- **PathImages** is the regular expression referring to the set of images whose metadata needs modifying.
- **Metadata** (named option) is the metadata to set (`F` for the focal length, `F35` for 35 mm equivalent focal length, `Cam` for the name of the camera model).

For example, to set the camera name to *UltraCam Eagle f210*:

```
mm3d SetExif ".*tif" Cam="UltraCam Eagle f210"
```

Here is an example of the end of the file *DicoCamera.xml* with a new camera, UltraCam Eagle f210:

```
<CameraEntry>
   <Name>  IQ180 </Name>
   <SzCaptMm> 53.7 40.4 </SzCaptMm>
   <ShortName> IQ180 </ShortName>
</CameraEntry>
```

[37] Another way to proceed is to write an XML file that describes all the data of your project. It shall be named *MicMac-LocalChantierDescripteur.xml.* It relies on specific keys that notify MicMac of the nature of the metadata (focal length, name of the camera…). If this file does not exist, MicMac will use the default settings. This tutorial does not require such file, but please feel free to read more about it in the documentation of MicMac.

[38] SetExif online documentation: https://micmac.ensg.eu/index.php/SetExif

```
    <CameraEntry>
       <Name>  UltraCam Eagle f210 </Name>
       <SzCaptMm> 104.052 68.016 </SzCaptMm>
       <ShortName> UCE-f210 </ShortName>
    </CameraEntry>

</MMCameraDataBase>
```

The focal length is probably missing from the data if the field of the focal length in the metadata of the photo is null (*right-click on the photo > Properties*):

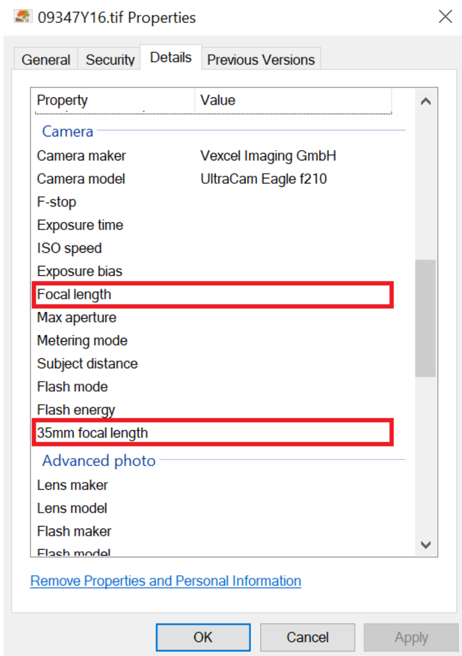


FIGURE 6: FOCAL LENGTH IN THE METADATA OF THE IMAGE

The following error informs that the focal length is unkown:

```
|    Sorry, the following FATAL ERROR happened
|
|     cMetaDataPhoto::Foc
```

To add the focal length (in millimetres) to the exif metadata of your images, you can use the MicMac command `SetExif`:

```
mm3d SetExif ".*tif" F=210.75
```

If you add the focal length and the dimensions of the sensor of the camera, you do not need to fill in the 35 mm equivalent focal length. It will be computed by MicMac.

As explained above, there is a second possibility. Instead of adding the focal length to the metadata and the dimensions of the sensor into the *DicoCamera.xml* file, you can add just the 35 mm equivalent focal length (in millimetres), using the MicMac command `SetExif`:

```
mm3d SetExif ".*tif" F35=95
```

II) Create the position file

You need the positions of the camera for each image in a text file to complete the absolute orientation.[39]

If the positions are GPS measurements in the metadata of the images, you can use the MicMac command XifGps2Txt to generate automatically the text file from the metadata:

```
mm3d XifGps2Txt PathImages
```

- **XifGps2Txt** [40] extracts the embedded information from the EXIF metadata and converts it into a text file.
- **PathImages** is the regular expression referring to the set of images.

For example:

```
mm3d XifGps2Txt ".*tif"
```

If the generated file is empty, it means that the position is not directly available in the EXIF metadata.

If the position is not directly available from metadata, you may write the text file yourself, in a format (content of the columns) you want and which you specify to MicMac when you use the command.

Whether the text file was generated by the XifGps2Txt tool or written by hand, when you use the tool OriConvert to convert your text file into the MicMac XML format, you have to specify the format (columns) of your file. You can do this either by directly using the descriptive string as a parameter in the command, or you can use the mode OriTxtInFile.[41] With this mode, the first line of the file of positions must be a comment line (the comment symbol is #) giving the format.
Note that MicMac has its own convention regarding column title:

TABLE 1: MICMAC CONVENTION FOR FILE FORMAT

Nature of the data	MicMac column name
Image name	N
X-coordinate or longitude	X
Y-coordinate or latitude	Y
Z-coordinate or altitude	Z
Uncertainty on X	Ix
Uncertainty on Y	Iy
Uncertainty on Z	Iz
Yaw (Kappa)	K
Pitch (Omega)	W
Roll (Phi)	P

These names define the format. For example, #F=N Y X Z K W P defines a file that has seven columns: name of the image, latitude, longitude, altitude, yaw, pitch, roll. If you give the format directly in the command line, you need an underscore instead of space between each column title, for example: "#F=N_Y_X_Z_K_W_P".

For instance if your file has four columns Name, X, Y and Z, you can either:

---

[39] We do not use ground control points (besides the camera centres) to georeference the model in this tutorial.
[40] XifGps2Txt online documentation: https://micmac.ensg.eu/index.php/XifGps2Txt
[41] Another mode exists, OriTxtAgiSoft, which is the orientation format of Agisoft PhotoScan.

- Use this command:

```
mm3D OriConvert "#F=N_X_Y_Z" GpsCoordinates.txt GPS_Georef
```

With this file (`GpsCoordinates.txt`):

```
IMG_0926.JPG 2.196733 48.859056 248.772003
IMG_0927.JPG 2.196122 48.859308 251.614000
IMG_0928.JPG 2.195386 48.859478 248.145997
```

The `XifGps2Txt` tool generates a file similar to this one.

- Or use this command :

```
mm3D OriConvert OriTxtInFile GpsCoordinates.txt GPS_Georef
```

With this file:

```
#F=N X Y Z
IMG_0926.JPG 2.196733 48.859056 248.772003
IMG_0927.JPG 2.196122 48.859308 251.614000
IMG_0928.JPG 2.195386 48.859478 248.145997
```

III) Define a projection system

Because of how MicMac is implemented, the coordinate system has to be Euclidian, or close to being Euclidian. In particular, you cannot work with the GPS coordinates in degrees (WGS84 system), because it is not Euclidian.

A common way to work around it is to work in the official local projection. It may not be precisely Euclidian (with significant alteration for large areas), but it is usually close enough for the model to be satisfying.

It is also possible to define a local Euclidian projection. MicMac can do that automatically. You work in a local tangent system (RTL[42]) defined around a 3D point (of the field) and its tangent plane.

You can transform your GPS coordinates into the (almost) Euclidian system using `OriConvert` and its optional parameter `ChSys`: [43]

```
mm3D OriConvert OriTxtInFile GpsCoordinates.txt GPS_Georef
ChSys=DegreeWGS84@Lambert72.xml
```

The syntax is **`ChSys=initial_coordinate_system@final_coordinate_system`**.

The systems are defined either by an XML file or by a keyword known by MicMac[44], such as:

- Geocentric Cartesian: GeoC (in command line), eTC_GeoCentr (in file)
- WGS84 geographic: WGS84 or DegreeWGS84 (in command line), eTC_WGS84 (in file)
- Local tangent system: eTC_RTL (in file)
- System described in proj4 format: eTC_Proj4 (in file)

If you use an official projection, you need to paste its description in the proj4 format [45] into an XML in the following format:

```
<SystemeCoord>
   <BSC>
      <TypeCoord>eTC_Proj4</TypeCoord>
      <AuxStr> # Your projection (here: Belgian Lambert 72, EPSG:31370)
+proj=lcc    +lat_1=51.16666723333333    +lat_2=49.8333339    +lat_0=90
+lon_0=4.367486666666666   +x_0=150000.013   +y_0=5400088.438   +ellps=intl
+towgs84=-106.869,52.2978,-103.724,0.3366,-0.457,1.8422,-1.2747  +units=m
+no_defs
      </AuxStr>
   </BSC>
</SystemeCoord>
```

The <TypeCoord> tag `eTC_Proj4` means that the following description is in the proj4 format. The <AuxStr> tag is for the string defining the projection.

If you want to work in a local tangent system, you have to use an XML in the following format:

---

[42] RTL: *Repère tangent local* in French.
[43] ChSys: Change System.
[44] See more about the XML format for coordinate system in the chapter *XML codage* of the MicMac pdf documentation.
[45] You can easily find online the proj4 definition associated with your projection, for instance on the website http://epsg.io/

```
<SystemeCoord>
   <BSC>
       <TypeCoord>eTC_RTL</TypeCoord>
          <AuxR>4320361.093</AuxR> #Centre of the RTL
          <AuxR>-98032.083</AuxR>
          <AuxR>4675380.773</AuxR>
   </BSC>
   <BSC>
      <TypeCoord>eTC_GeoCentr</TypeCoord> #Coordinate system of the centre
   </BSC>
</SystemeCoord>
```

The `<TypeCoord>` tag `eTC_RTL` means that the following description is a local tangent system. The `<AuxR>` tag is for a double, one coordinate of the centre of the projection. The `<TypeCoord>` tag `eTC_GeoCentr` is the coordinate system of the centre.

NB: At the time this tutorial is written, **the *ChSys* command to change coordinate system might not be working correctly**,[46] due to MicMac still being under development. MicMac developers recommend using MicMac on Linux, or use another software to project coordinates instead (QGIS, proj4…), and give to MicMac already projected positions in the text file of positions.

We recommend you always check the transformed coordinates of the centres in the XML files generated by MicMac before you compute the absolute orientation.

The centre (position of the camera) of each image is stored in a file named *Orientation-[image_name].xml* in the orientation directory (e.g. *Ori-GPS_Georef*), where it is marked by a `<Centre>` tag. Here is an example of coordinates for centre in such a file:

```
<OrientationConique>
     …
     <Externe>
          <Time>0</Time>
          <KnownConv>eConvApero_DistM2C</KnownConv>
          <Centre>
          151302.369999999995 168062.410000000003 3111.19000000000005
          </Centre>
```

If you cannot trust the MicMac transformation, transform the coordinates into the (almost) Euclidian system you want with an external tool (GIS such as QGIS typically), and **write directly the projected coordinates in the position text file.** [47]

You can either run (e.g.):

```
mm3D OriConvert "#F=N_X_Y_Z" GpsCoordinates.txt GPS_Georef
ChSys=DegreeWGS84@Lambert72.xml
```

where *GpsCoordinates.txt* is in WGS84 and is transformed into Belgian Lambert 72 by the `OriConvert` tool;

[46] Both successful and failed transformations have been experienced, depending on the initial and final coordinate systems.
[47] The position text file is described in the chapter *How to add the needed metadata, II) Create the position file.*

or run:

```
mm3D OriConvert "#F=N_X_Y_Z" GpsCoordinates.txt GPS_Georef
```

where *GpsCoordinates.txt* is already in Belgian Lambert 72.

How to choose parameter values

I) XML file of pairs of images

With the `OriConvert` tool, it is possible to generate an XML file that stores couples of images that are likely overlapping, given their respective positions. MicMac will try to match extracted features only in the listed couples of images. If the photographed area is large and each image only overlaps with a small proportion of images of the dataset, using such file can save processing time.

To illustrate the impact of this parameter, the processing time of two steps of the process chain was measured for two datasets. Each dataset was tested on subsets of 2, 5, 10, 15, 20 and 24 images (and 28 for set 2).

The two steps are:

- *OriConvert* (first line): generate the XML file.
- *Tapioca* (second line): compute tie points using (or not) the XML file.

The rest of the chain is not impacted by this parameter (apart from the fact that computed tie points differ).

Two sets are tested:

- *Set 1* (column 1) is made up of 24 tif aerial photographs, each weighs 770 MB.[48]
- *Set 2* (column 2) is made up of 28 JPG aerial photographs, each weighs around 3 MB.

The two same commands were run for both sets:

With file:

```
mm3d OriConvert "#F=N_X_Y_Z" GpsCoordinatesFromExif.txt Georef
NameCple=Couples.xml

mm3d Tapioca File "Couples.xml" 2000
```

Without file:

```
mm3d OriConvert "#F=N_X_Y_Z" GpsCoordinatesFromExif.txt Georef

mm3d Tapioca All ".*tif" 2000 # ".*JPG" for set 2
```

[48] The images are part of the Brussels UrbIS ®© and were provided graciously by the cirb.brussels.

Here are the obtained results:

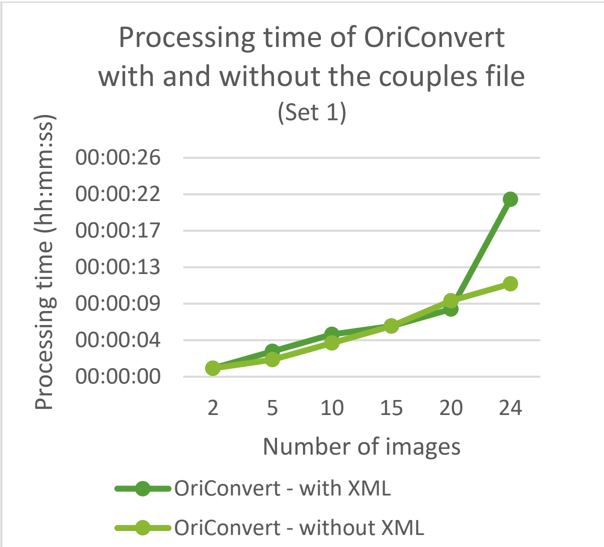


FIGURE 7: PROCESSING TIME OF ORICONVERT FOR DIFFERENT NUMBERS OF IMAGES WITH AND WITHOUT FILE OF COUPLES (SET 1)

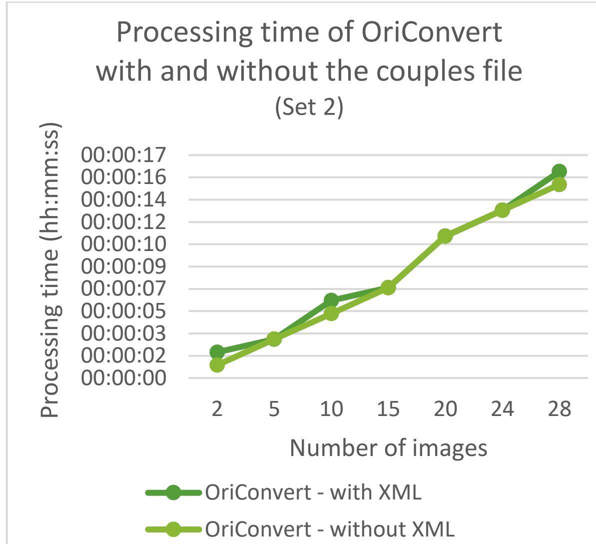


FIGURE 8: PROCESSING TIME OF ORICONVERT FOR DIFFERENT NUMBERS OF IMAGES WITH AND WITHOUT FILE OF COUPLES (SET 2)

`OriConvert` computes the XML file. Therefore, this step takes more time with the option "with file". However, the difference is not significant.

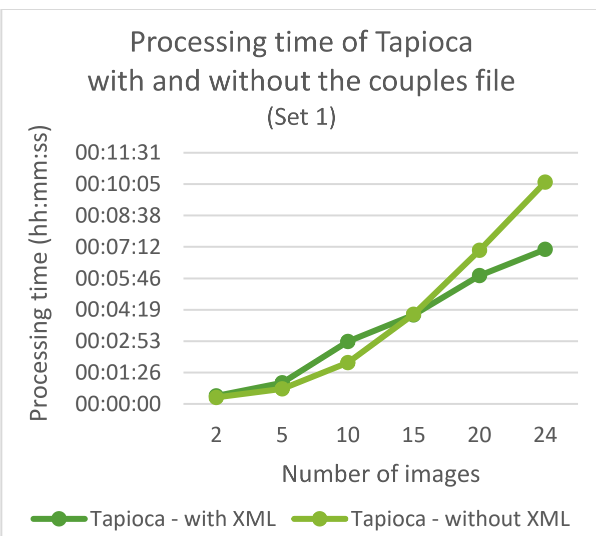


FIGURE 9: PROCESSING TIME OF TAPIOCA FOR DIFFERENT NUMBERS OF IMAGES WITH AND WITHOUT FILE OF COUPLES (SET 1)

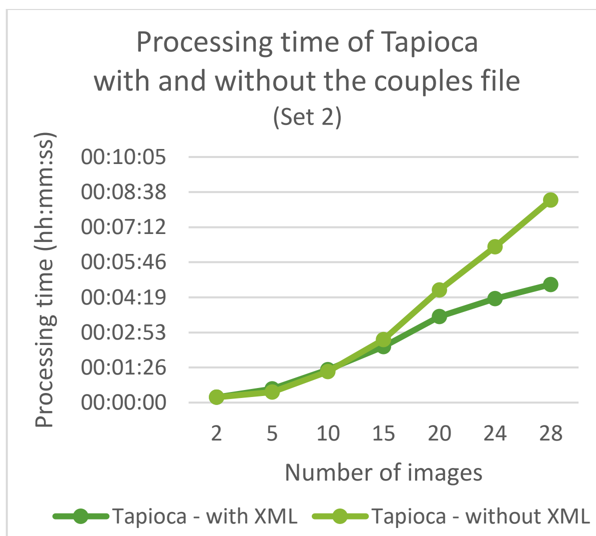


FIGURE 10: PROCESSING TIME OF TAPIOCA FOR DIFFERENT NUMBERS OF IMAGES WITH AND WITHOUT FILE OF COUPLES (SET 2)

`Tapioca` computes tie points. With the XML file of couples, fewer couples are tested. Therefore, this step is faster with the option "with file". The more images, the more time you save with the file. With less than 15 images, the file does not save time.

> For a dataset made up of under 15 images, the XML file of pairs is useless but the additional amount of time spent with this mode is not significantly large (a few seconds).
> For over 15 images, the XML file saves time.

When it comes to processing time, the XML file is profitable. However, some couples might be missed with the file, which would be matched without the option.

The following table illustrates this loss of information. Only *Set 2* was tested.

- Number of images: number of images of the tested subset
- With XML file of pairs: Y for Yes (option with file), N for No (without the file)
- Number of configurations: a configuration is a subset of images that share multiple tie points[49]. For instance, if there are tie points between image 1 and 2 (not detected on image 3), and multiple tie points on image 1, 2 and 3 (meaning that they are tied by common details), there are two configurations: images 1-2 and images 1-2-3. A stereo pair necessarily has one configuration. A configuration is similar to a couple, but defined for over two images. The number of configurations detected is therefore dependent on the XML file. For instance, if image 3 is not matched with image 1 and image 2 in the XML file, configuration 1-2-3 will not be detected.
- The number of tie points: it is the number of vertices in the sparse point cloud computed by `AperiCloud`. Multiple tie points are counted multiple times. For instance, a tie point between images 1, 2 and 3 will be counted three times: once per couple 1-2, 2-3 and 1-3.
- Percentage of tie points detected with XML file: it is the ratio (as a percentage) of the number of tie points detected with file over the number of tie points detected without. A 100 % percentage means that not using the XML file does not increase the number of detected tie points.

TABLE 2: QUALITY INDICATORS OF THE COMPUTATION OF TIE POINTS WITH AND WITHOUT FILE OF COUPLES (SET 2)

Number of images	2		5		10		15		20		24		28	
With XML file of pairs	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Number of configurations	1	1	26	26	487	493	1616	1706	1067	3815	4441	5054	5891	7062
Number of tie points	3303	3303	14162	14162	58346	58602	85027	86482	127127	129369	150138	153261	180667	186245
Percentage of tie points detected with XML file	100 %		100 %		99.56 %		98.32 %		98.27 %		97.96 %		97.01 %	

> You do not miss a significant number of tie points when you limit the search of tie points to the list of couples stored in the XML.
> The 3D models and DSMs generated from a same dataset with and without the XML file of couples look similar.

> By using the XML file of pairs, you save processing time and do not decrease significantly the quality of the results of the computation of tie points.
> **When in doubt, you should use this option.**

[49] A multiple tie point is a tie point visible on more than two images.

II) Resolution for computation of tie points

When you use `Tapioca`, you can choose the resolution at which the tie points are computed. It is the number of pixels of the larger side of your image when subsampled.

For example (resolution of 2000 pixels):

```
mm3d Tapioca File Couples.xml 2000
```

The processing time of each step has been collected for a set of 24 aerial images of Brussels (770 MB each, 20010x13080 px, pixel size 7.3 m).[50] These results may give you an idea of what the processing time of each step might be, according to the resolution of the subsampling, and in particular how much time it might take to increase the resolution.

`C3DC` and `Pims2MNT` were computed in `MicMac` mode.

Here is the full set of commands used:

```
mm3d OriConvert "#F=N_X_Y_Z" CoordsGPS.txt Georef NameCple=Couples.xml

mm3d Tapioca File Couples.xml 6000 # Here is the resolution

mm3d Tapas RadialStd ".*tif" Out=Relative
mm3d CenterBascule ".*tif" Relative Georef Bascule
mm3d Campari ".*tif" Bascule Compense EmGPS=[Georef,2] AllFree=1

mm3d AperiCloud ".*tif" Compense
mm3d C3DC MicMac ".*tif" Compense
mm3d Pims2MNT MicMac
```

`Tapioca` was used in `File` mode with the resolutions 2000 px, 6000 px and 10000 px; on subsets of 5, 12 and 24 images.

`Tapioca` was then tested in `MulScale` mode. This mode uses two resolutions. A first computation of tie points is made for all the pairs of images at very low resolution (faster), then the actual computation, at the high resolution, is done only for the pairs having, at low resolution, a number of tie points exceeding a given threshold.

The `MulScale` mode was tested on the full set of 24 images with resolutions 2000 px (low) and 6000 px (high), 2000 px and 10000 px, and 6000 px and 10000 px. The rest of the chain remains unchanged, apart from `OriConvert` that does not produce the XML file of couples anymore as you cannot combine the `MulScale` and the `File` mode.

---

[50] The images are part of the Brussels UrbIS ®© and were provided graciously by the cirb.brussels.

Here are the processing times[51] for each resolution (in hh:mm:ss), for every step of the process chain, and for different sizes of subsets:

TABLE 3: PROCESSING TIME OF EACH STEP OF THE PROCESSING CHAIN FOR DIFFERENT RESOLUTIONS OF TAPIOCA AND DIFFERENT NUMBERS OF IMAGES

Number of images	5			12			24					
Tapioca Resolution	2000	6000	10000	2000	6000	10000	2000	6000	10000	2000 6000	2000 10000	6000 10000
OriConvert	0:00:02	0:00:03	0:00:03	0:00:09	0:00:07	0:00:09	0:00:12	0:00:53	0:00:18	0:00:15	0:00:19	0:00:19
Tapioca	0:01:04	0:05:25	0:17:00	0:03:31	0:21:27	1:08:51	0:08:26	0:55:35	3:22:14	1:39:47	5:38:58	5:18:49
Tapas	0:00:49	0:03:26	0:07:53	0:02:50	0:16:42	0:36:36	0:05:44	0:21:34	0:45:52	0:21:41	0:46:57	0:46:22
CenterBascule	0:00:00	0:00:00	0:00:02	0:00:01	0:00:01	0:00:01	0:00:01	0:00:11	0:00:01	0:00:01	0:00:01	0:00:01
Campari	0:00:14	0:01:00	0:02:16	0:00:46	0:03:15	0:07:36	0:00:02	0:05:20	0:16:26	0:07:26	0:11:55	0:16:26
AperiCloud	0:00:20	0:00:20	0:00:28	0:00:39	0:00:54	0:01:21	0:01:18	0:01:56	0:02:32	0:01:40	0:03:00	0:03:06
C3DC	0:11:35	0:12:14	0:13:18	0:24:32	1:48:09	0:29:35	0:20:39	0:42:52	1:27:11	0:43:09	0:41:43	0:40:41
Pims2MNT	0:02:04	0:02:11	0:02:12	0:07:48	0:05:27	0:06:01	0:05:42	0:18:03	0:11:19	0:14:27	0:11:31	0:09:19
TOTAL	0:16:08	0:24:41	0:43:12	0:40:16	2:36:03	2:30:11	0:42:04	2:26:27	6:05:54	3:08:26	7:34:25	7:15:04

The three following graphs show the processing times for 5, 12 and 24 images, for each resolution, for different steps of the processing chain: `Tapioca` (whose resolution differs), `Tapas` (the step immediately following that depends on the resolution of `Tapioca` the most), and the total processing time of the chain. For the `MulScale` mode (resolutions of 2000 (low) - 6000 (high), 2000 - 10000 and 6000 - 10000) only the full set of 24 images was tested.[52] Therefore, you can only see one point (instead of a full line).

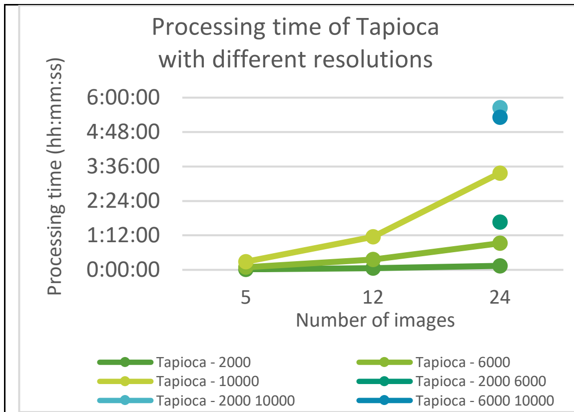


FIGURE 11: PROCESSING TIME OF TAPIOCA FOR DIFFERENT RESOLUTIONS AND NUMBERS OF IMAGES

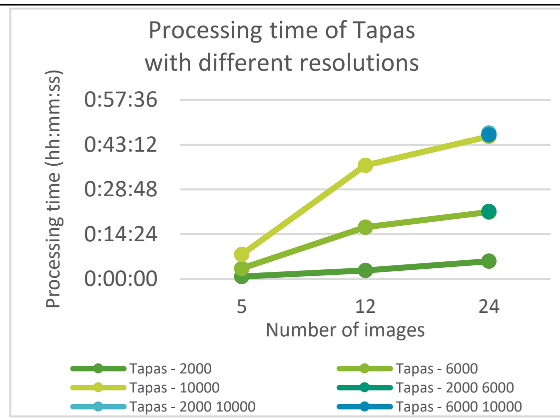


FIGURE 12: PROCESSING TIME OF TAPAS FOR DIFFERENT RESOLUTIONS OF TAPIOCA AND DIFFERENT NUMBERS OF IMAGES

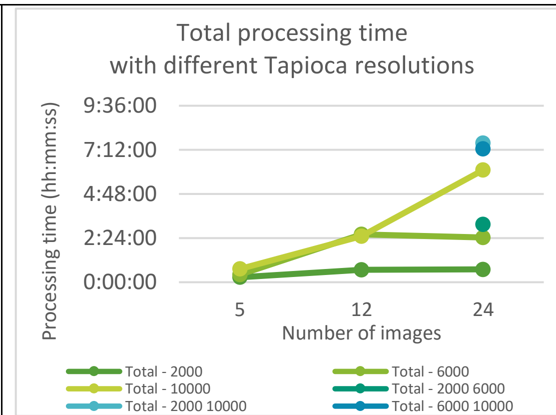


FIGURE 13: TOTAL PROCESSING TIME OF THE CHAIN FOR DIFFERENT RESOLUTIONS OF TAPIOCA AND DIFFERENT NUMBERS OF IMAGES

[51] Please note that these values were measured once, and are not an average result. Therefore, they hold high uncertainty. Only their order of magnitude is meaningful.
[52] Testing this mode takes hours of processing. You can see the evolution of time according to the number of images with the other modes. The tests on the set of 24 images with the `MulScale` mode gives you an idea of the value for this mode for the full dataset for comparison.

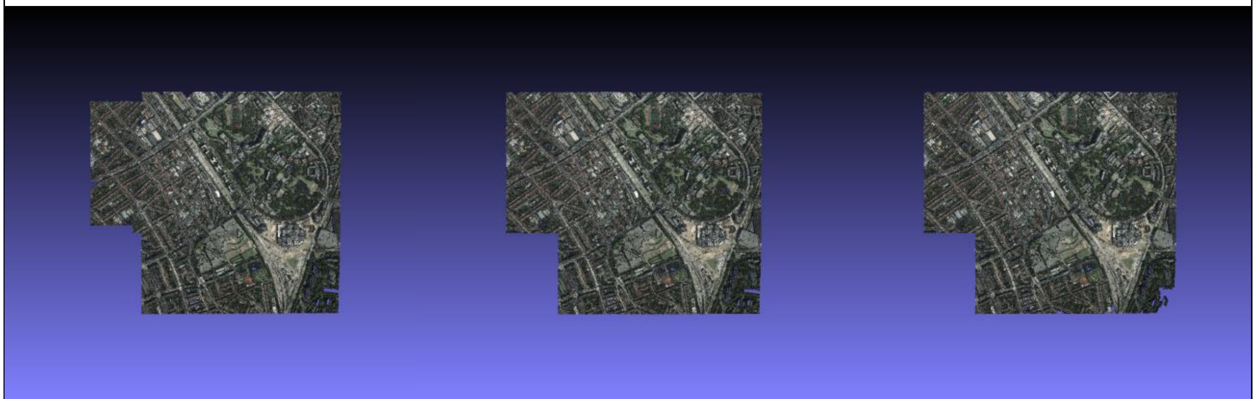
The choice of resolution obviously has an impact on quality, which differs from a dataset to another. Here are some quality indicators for each resolution for the tested dataset. Please keep in mind that they illustrate the behaviour of MicMac, but you may obtain better results with other settings for your specific dataset.

TABLE 4: QUALITY INDICATORS OF THE FINAL PRODUCTS FOR DIFFERENT RESOLUTIONS OF TAPIOCA AND DIFFERENT NUMBERS OF IMAGES

Number of images			5			12			24					
Tapioca resolution			2000	6000	10000	2000	6000	10000	2000	6000	10000	2000 6000	2000 10000	6000 10000
MicMac Tool	Indicator	Best when												
Tapioca	Average number of tie points detected per image	High	12155.6	55270.4	125265	16332.83	70939.25	158289.83	15451.54	61992	134921.92	62148.96	135392	135389.67
Tapas	Average residual (px)	Low (< 1)	1.36	0.67	0.45	1.57	0.78	0.53	1.6	0.82	0.55	0.82	0.55	0.55
	Worst residual (px)	Low (<1)	1.51	0.8	0.55	1.65	0.84	0.58	1.72	0.91	0.62	0.91	0.62	0.62
	Average percentage of points kept for computation (%)	High (> 95)	94.52	99.15	99.69	92.96	98.87	99.6	92.26	98.68	99.52	98.66	99.5	99.51
	Worst percentage of points kept for computation (%)	High (> 95)	92.52	98.7	99.51	92.13	98.63	99.47	89.13	97.92	99.23	97.9	99.21	99.21
CenterBascule	Average residual (distance, metre)	Low (5-10)	7.62	6.09	6	10.21	10.19	10.15	8.81	9.68	9.74	9.69	9.77	9.77
	Worst residual (distance, metre)	Low (5-10)	13.33	10.79	10.58	14.2	14.34	14.31	13.95	14.55	14.52	14.58	14.55	14.55
Campari	Average residual (px)	Low (< 1)	1.22	0.66	0.65	1.44	0.77	0.6	1.52	0.99	0.74	1.21	0.65	0.65
	Worst residual (px)	Low (< 1)	1.41	0.82	0.67	1.52	0.85	0.69	2.28	2.23	1.16	2.26	0.9	0.9
	Average percentage of points kept for computation (%)	High (> 95)	94.36	99.1	96.74	92.79	98.83	99.58	87.17	93.16	90.54	93.29	89.28	89.29
	Worst percentage of points kept for computation (%)	High (> 95)	92.04	98.58	95.52	91.71	98.47	99.37	21.36	19.76	51.04	22.45	51.08	51.09
AperiCloud	Number of points	High	68327	284073	633938	212246	868506	1916630	390761	1475745	3178834	1481187	3146461	3146846
C3DC	Number of points	High	24275597	24621896	24419099	46225157	46818977	46963586	81784418	81433996	70854532	80870168	70006351	70064360
Georeferencement: order of magnitude on the estimated position error (m)	X	Low (< 5m)	2	2	2	2	3	3	5	5	2	5	2	2
	Y	Low (< 5m)	30	15	12	10	10	11	10	13	4	12	5	5
	Z [53]	Low (< 5m)	555	405	335	125	105	110	575	635	15	625	20	20

[53] The products have an obvious shift in Z. Distance from the camera was poorly estimated. Relative heights remain satisfying.

Finally, here is how the computed 3D point cloud look like for each resolution:

5 images		
2000	6000	10000
		
12 images		
2000	6000	10000
		
24 images		
2000	6000	10000
		

24 images		
2000 6000	2000 10000	6000 10000

FIGURE 14: 3D POINT CLOUD PRODUCED FOR DIFFERENT RESOLUTIONS OF TAPIOCA AND FOR DIFFERENT NUMBERS OF IMAGES (SNAPSHOTS ON MESHLAB)

Increasing the resolution of computation of tie points significantly increases the processing time.

With a higher resolution, more points are detected. During bundle adjustments, residuals improve because more points are dropped.

It results in a point cloud denser in areas that could be modelled, but with potentially large holes with no point where orientation is not as satisfying as for the rest of the area. The total number of points in the clouds increases (in general) with the resolution, but they densify well-modelled areas and desert poorly orientated areas. Residuals are better in higher resolutions because only parts whose orientations are very consistent are kept. **However, with higher resolution the model is not necessarily better. It may be more incomplete.**

Note that the larger the area, the higher the incoherence of orientation might be. Therefore, smaller areas do not have as many holes as wide areas. For example with 5 images, the cloud is more homogeneous in high resolution than with 24 images.

> **The computation of tie points in high resolution increases the processing time and does not necessarily lead to a better model.** It densifies areas whose orientations are remarkably coherent with one another, whilst in the contrary other areas are sparse.
> **You should use higher resolution when the orientation is very accurate and you need a denser model.**
> **You should try and lower the resolution when the point cloud has holes.**

III) Choose a model of calibration

The coefficients of calibration (values specific to each camera-lens pair and the settings for the acquisition) are estimated by least squares adjustment with distortion measurements taken from a set of images (taken with the same camera and unchanged settings). The adjustment algorithm computes the coefficients of the model of distortion chosen by the user based on the distortion measurements of the dataset. For the estimation to be optimal, the images must vary the angles of view, the distance from the photographed details, the orientation… Same details should be photographed from very different angles to have more meaningful measurements of distortion.

Calibration models

Many distortion models are available in MicMac. We recommend you read the MicMac documentation for full details (mathematical equations of the models in particular).[54] Here are some of the most common:

- **RadialExtended**: model with radial distortion; with 10 degrees of freedom: 1 for focal length, 2 for principal point, 2 for distortion centre, 5 for coefficients of radial distortion (r3, r5 . . . r11);
- **RadialStd**: a "subset" of previous model; radial distortion with 8 degrees of freedom: 1 for focal length, 2 for principal point, 2 for distortion centre, 3 for coefficients of radial distortion (r3, r5, r7);
- **RadialBasic**: a "subset" of previous model; radial distortion with 5 degrees of freedom : 1 for focal length, 2 for principal point and distortion centre (they are constrained to have the same value), 2 for coefficients of radial distortion (r3 and r5);
- **Fraser**: radial model, with decentric and affine parameters[55]; there are 12 degrees of freedom: 1 for focal length, 2 for principal point, 2 for distortion centre, 3 for coefficients of radial distortion (r3, r5 r7), 2 for decentric parameters, 2 for affine parameters; the optional parameters LibAff and LibDec (def value true) can be set to false if decentric or affine parameters must remain frozen;
- **FraserBasic**: a "subset" of previous model: 10 degrees of freedom, same as previous with principal point and distortion center constrained to have the same value;
- Models for **fisheye** lenses;
- Polynomial models;
- **AutoCal** and **Figee:** with these modes no model is defined, all the calibration must have a value (via `InCal` or `InOri` options); with `AutoCal` the coefficients of calibration are re-evaluated while with `Figee` they are frozen. These modes are useful if you want to reuse the orientation computed using a subset of images for a larger set.

> **When in doubt, we recommend you use `RadialStd`.** It provides satisfying results almost systematically. We recommend you try it, and change if needed according to the results you obtained with this model.

> The computed coefficients of calibration are stored in the orientation directory, in the file named ***AutoCal_Foc-[focal_length_in_μm]_Cam-[name_of_the_camera].xml***.

---

[54] Tapas online documentation with the list of models of distortion: https://micmac.ensg.eu/index.php/Tapas
Please find details in the pdf documentation: which parameters are estimated for each model, and what are the mathematical equations associated with the models. See the *distortion models* in the chapter *Simple relative orientation and calibration with Tapas* of the chapter *Simplified Tools* (chapter 3.4.3.2 p. 52 in the version of April 20, 2018).
[55] The main difference between Fraser and Radial models is that Fraser has an affine and a decentric correction.

How to assess the quality of a model

You want to use a model with as few parameters as possible because:

- You do not want to "over-parametrize" the estimation. You want to estimate what is necessary.
- Distortion parameters correlate strongly with one another and with parameters of the external orientation.
- There is a risk of divergence with models with numerous parameters.

You may want to use a model with numerous parameters if:

- Distortion models with fewer parameters do not allow modelling your data with enough precision.
- Distortion models with numerous parameters converge and improve the accuracy of the final products.

Here are some ways to tell whether the model of distortion is appropriate for your data:

- For steps of bundle adjustment (`Tapas`, `Campari`), the error of re-projection (residual, *ER2* in the terminal)[56] is low (< 1 px in general). It means that the estimated model is internally coherent, which may not happen if the model of distortion is not appropriate. However, please note that you may have small residuals with an inaccurate final 3D model (especially for some configurations of acquisition such as linearly). Residuals only give information about the internal coherence. You need to use ground points to assess the accuracy of the absolute orientation.
- You can use the tool `MMTestOrient`.[57] It compares a secondary image to a master image and generates two images: Px1 (geometry: relative depth) and Px2 (parallax, the remaining error after the estimated calibration is applied). If you notice an important systematic error on Px2, you may want to try again with another model of distortion (free more parameters of calibration or use a more complex model of distortion).

  ```
  mm3d MMTestOrient Image1 Image2 Orientation_directory
  ```

Reuse a model of calibration

You can use a subset to compute a first orientation, which you can then reuse on a larger set. If the larger set includes images that are not well linked to the others, the orientation computed using the whole set may be inaccurate. Using a subset of images with a better configuration (large overlapping, homogenous lighting, texture, multiple tie points…) to compute the model may help.

Firstly, run `Tapas` on a subset of images, for example:

```
mm3d Tapas RadialStd "img[0-7].tif" Out=Calib
```

Then, use the calibration obtained on this subset for the whole dataset using the mode `AutoCal` (meaning that the calibration is based on the `InCal` parameter) and the parameter `InCal` (which is the orientation directory you want to reuse), for example:

```
mm3d Tapas AutoCal ".*tif" InCal=Calib Out=Relative
```

---

[56] See *Step 3 – Calibration of the camera and relative orientation* about `Tapas` for more details about the residuals.
[57] Please refer to the documentation in the terminal for more details (`mm3d MMTestOrient -help`), or to the pdf documentation (chapter *Use cases for 2D Matching, Checking orientation, For Conik Orientation*; 12.1.1 p. 215 in the version of April 20, 2018).

Only the parameters of calibration will be reused.

If you want to reuse all the orientations, you can use similarly the parameter `InOri`; for example:

```
mm3d Tapas RadialStd "img[0-7].tif" Out=Subset
mm3d Tapas AutoCal ".*tif" InOri=Subset Out=Relative
```

It enables the user to give more weight to the subset on which the first orientation is calculated. It is useful for instance if you want to add an image that is not well tied to the others and could alter the overall relative orientation, or simply if you have already processed a dataset that you then want to complete with additional images without having to start over the whole process.

It is also necessary if you want to work on a pair of images, as the estimation of the calibration generally fails, and the transformation into the absolute system cannot be done without GCPs (in addition to the positions of the camera).

> As for now, it seems not possible to use directly calibration values from a calibration report. However, if you already know the coefficients of distortions of one of the models implemented by MicMac, you can modify (or write) the MicMac XML file of calibration by hand. If you want to do so, please make sure that the coefficients you have and the coefficients that MicMac stores in the XML are the same (same variables of the same mathematical equations).

IV) C3DC mode

Dense correlation with C3DC

The `C3DC` command computes a coloured, orientated point cloud from a set of oriented images.

`C3DC` uses masks (created using tie points) and depth maps, which are raster images that give the distance between the camera and the details of the photographs (they are stored in the generated directory *PIMs-[name_of_the_mode]*[58]).

You can run `C3DC` with different modes.

MicMac works with scale-space pyramids (directory *Pyram*). It starts the correlation on images with reduced resolution, and then successfully improves the resolution. `ZoomF` is the final zoom: 1 is full resolution, 2 is half... Highest resolutions take much longer to process, and do not necessarily improve the quality of the point cloud.

The main modes are:

- **QuickMac**: multistereoscopic with ZoomF=8, i.e. low preset 1pt/64px
- **MicMac**: multistereoscopic with ZoomF=4, i.e. medium preset 1pt/16px
- **BigMac**: multistereoscopic with ZoomF=2 i.e. high preset 1pt/4px

But you can also use:

- **Statue**: stereoscopic (strictly epipolar mode (i.e. best stereographic pairs)) with ZoomF=2, i.e. high preset 1pt/4px
- **Forest**: stereoscopic (strictly epipolar mode (i.e. best stereographic pairs)) with ZoomF=4, i.e. medium preset 1pt/16px

The epipolar geometry approach of these two modes should be better but slower.

> **When in doubt, we recommend you use `MicMac`.** It usually produces satisfying results.
> You can try other modes if the result with `MicMac` is not satisfying.

The three main modes `QuickMac`, `MicMac` and `BigMac` were tested for a set of 24 aerial images of Brussels (770 MB each, 20010x13080 px, pixel size 7.3 m, 2016).[59] The correlation with `C3DC` was made using the exact same data: the files processed in previous steps of the chain are the same for each subset. `Tapioca` was used in `File` mode with a resolution of 2000 px. Tapas was used in `RadialStd` mode.

The full chain used is:

```
mm3d OriConvert "#F=N_X_Y_Z" CoordsGPS.txt Georef NameCple=Couples.xml

mm3d Tapioca File Couples.xml 2000

mm3d Tapas RadialStd ".*tif" Out=Relative
mm3d CenterBascule ".*tif" Relative Georef Bascule
mm3d Campari ".*tif" Bascule Compense EmGPS=[Georef,2] AllFree=1

mm3d AperiCloud ".*tif" Compense
```

---

[58] PIMs: Per Image Matchings tool, called by C3DC for correlation.
[59] The images are part of the Brussels UrbIS ®© and were provided graciously by the cirb.brussels.

Three subsets of 5, 12 and 24 images were tested. These first steps were run three times, once for each subset.

The files at the end of this process were saved and the tested steps were run using these files.

```
mm3d C3DC MicMac ".*tif" Compense #Or QuickMac, BigMac

mm3d Pims2MNT MicMac #Or QuickMac, BigMac
```

Here are the sparse point clouds of tie points obtained with `AperiCloud`, before the tested steps:

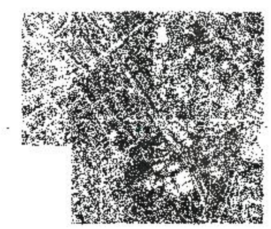
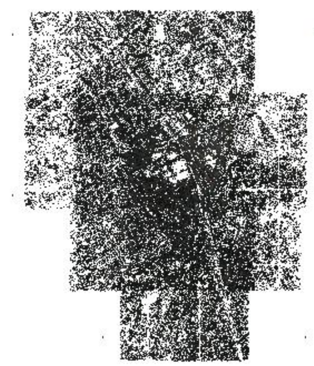
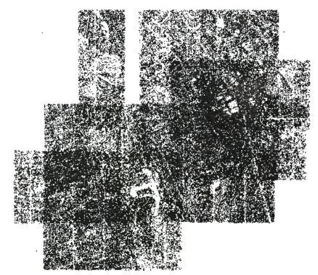
Number of images	5	12	24
Point cloud (snapshots on MeshLab)			
Number of points	68327	212246	390761

FIGURE 15: SPARSE POINT CLOUD (APERICLOUD) FOR DIFFERENT NUMBERS OF IMAGES

Processing time was measured for subsets of 5, 12 and 24 images.

TABLE 5: PROCESSING TIME (HH:MM:SS) OF C3DC FOR THREE MODES AND DIFFERENT NUMBERS OF IMAGES

Number of images	QuickMac	MicMac	BigMac
5 images	00:03:22	0:11:35	00:45:40
12 images	00:07:22	0:24:32	01:24:44
24 images	00:13:10	0:20:39	02:16:14

Naturally, the better the final resolution, the longer it takes to produce the point cloud.

The choice of resolution obviously has an impact on quality, which differs from a dataset to another. Here are some quality indicators for each mode for the tested dataset. Please keep in mind that they illustrate the behaviour of MicMac, but you may obtain better results with other settings for your specific dataset.

The following table shows the number of points for each cloud:

TABLE 6: NUMBER OF POINTS IN DENSE POINT CLOUD FOR DIFFERENT C3DC MODES AND DIFFERENT NUMBERS OF IMAGES

Number of images	QuickMac	MicMac	BigMac
5 images	6538412	24275597	79588859
12 images	12754453	46225157	130983428
24 images	23379879	81784418	202520142

Here are the point clouds computed for each mode and number of images:

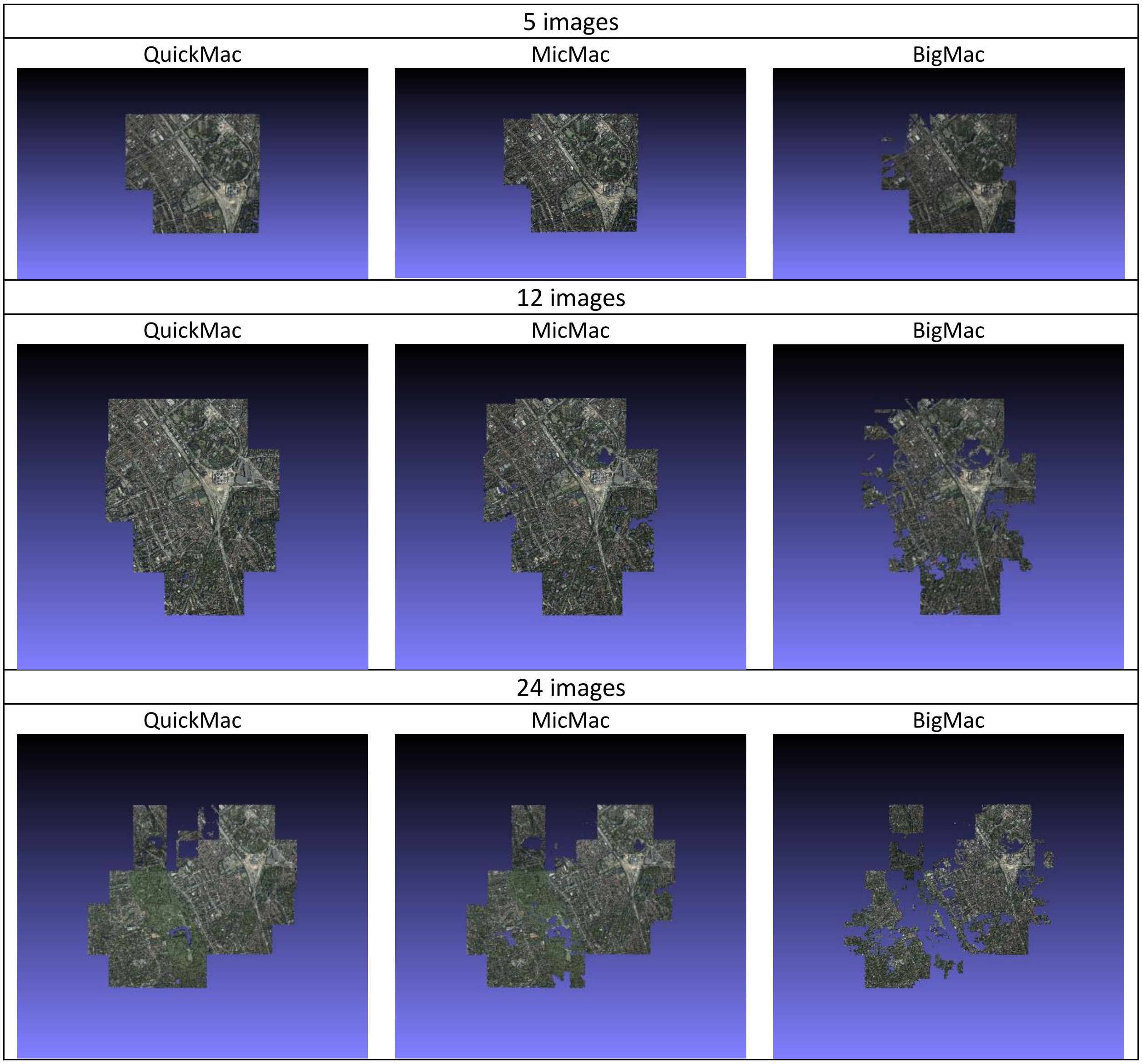


FIGURE 16: POINT CLOUDS FOR DIFFERENT C3DC MODES AND DIFFERENT NUMBERS OF IMAGES (SNAPSHOTS ON MESHLAB)

The results are similar to the tests on `Tapioca`. When you improve the resolution, well-modelled areas are densified, whilst the others are dropped.

> **The dense correlation in high resolution increases the processing time and does not necessarily lead to a better model.** It densifies areas whose orientations are remarkably coherent with one another, whilst in the contrary other areas are sparse.
> **You should use higher resolution when the orientation is very accurate and you need a denser model.**
> **You should try and lower the resolution when the point cloud has holes.**

Production of a DSM with Pims2MNT

When you use `Pims2MNT` after `C3DC`, the DSM is produced reusing the depth maps generated by `C3DC`. Therefore, the mode you choose for `C3DC` also has a direct impact on the DSM produced with `Pims2MNT`. When you run the command, you need to indicate which `C3DC` mode was used so that `Pims2MNT` will use the files in the directory *PIMs-[name_of_the_C3DC_mode]* for computation.

To study the impact of the mode of C3DC on the DSM, the processing time of `Pims2MNT` was measured for the same subsets of 5, 12 and 24 images, after the `C3DC` correlation.

TABLE 7: PROCESSING TIME (HH:MM:SS) OF PIMS2MNT FOR THREE MODES AND DIFFERENT NUMBERS OF IMAGES

Number of images	QuickMac	MicMac	BigMac
5 images	00:02:03	0:02:04	00:02:23
12 images	00:10:26	0:07:48	00:08:46
24 images	00:13:09	0:05:42	00:09:45

The `C3DC` mode has no significant impact on the processing time of `Pims2MNT` for these subsets.

The produced DSM all have a pixel size of 0.15 m, whatever the mode (it can be changed with parameters of the `Pims2MNT` command; default settings were used here).

Here are the obtained DSM:

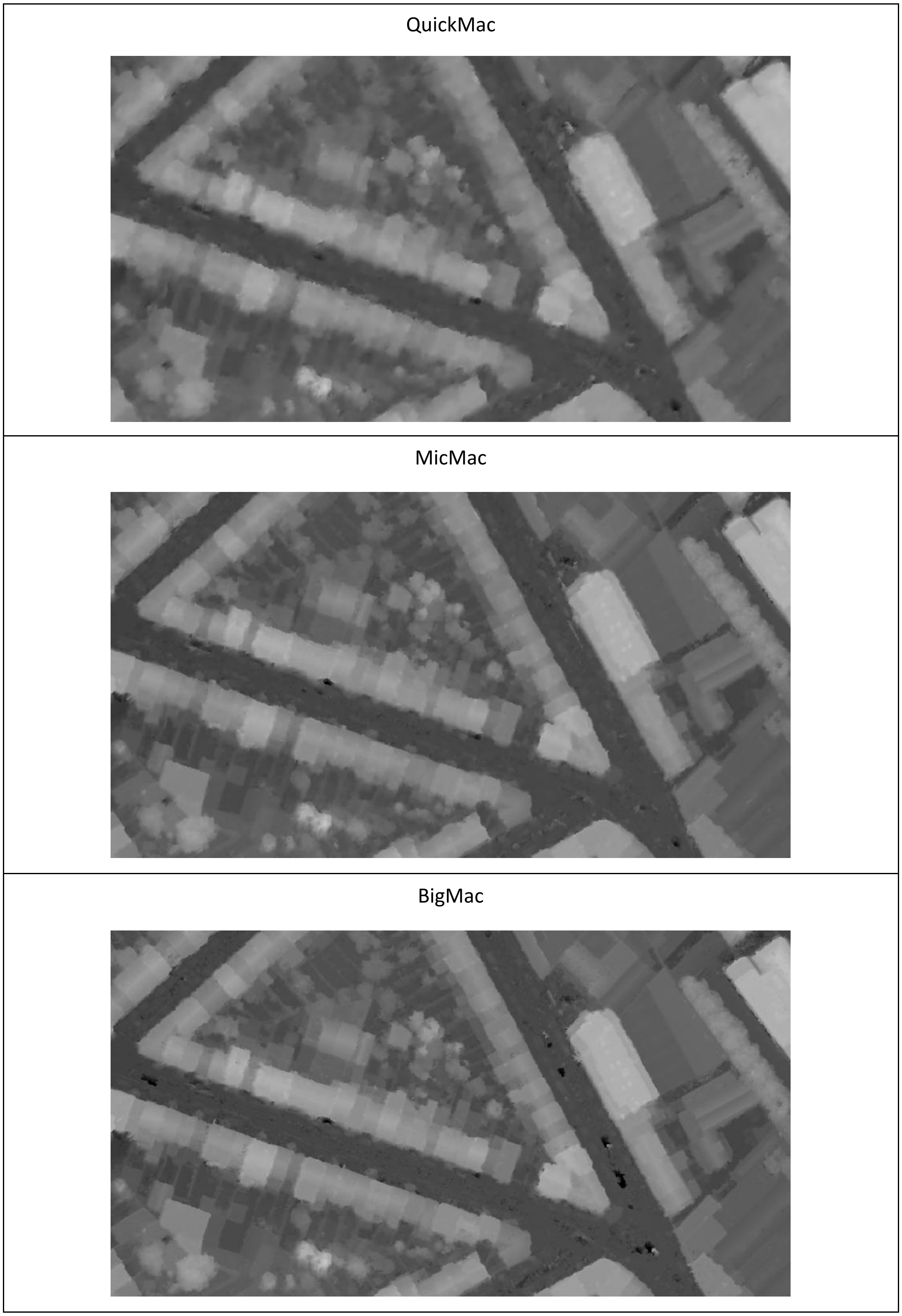


FIGURE 17: DSM (DETAIL) FOR DIFFERENT C3DC MODES FOR 5 IMAGES (FROM QGIS)

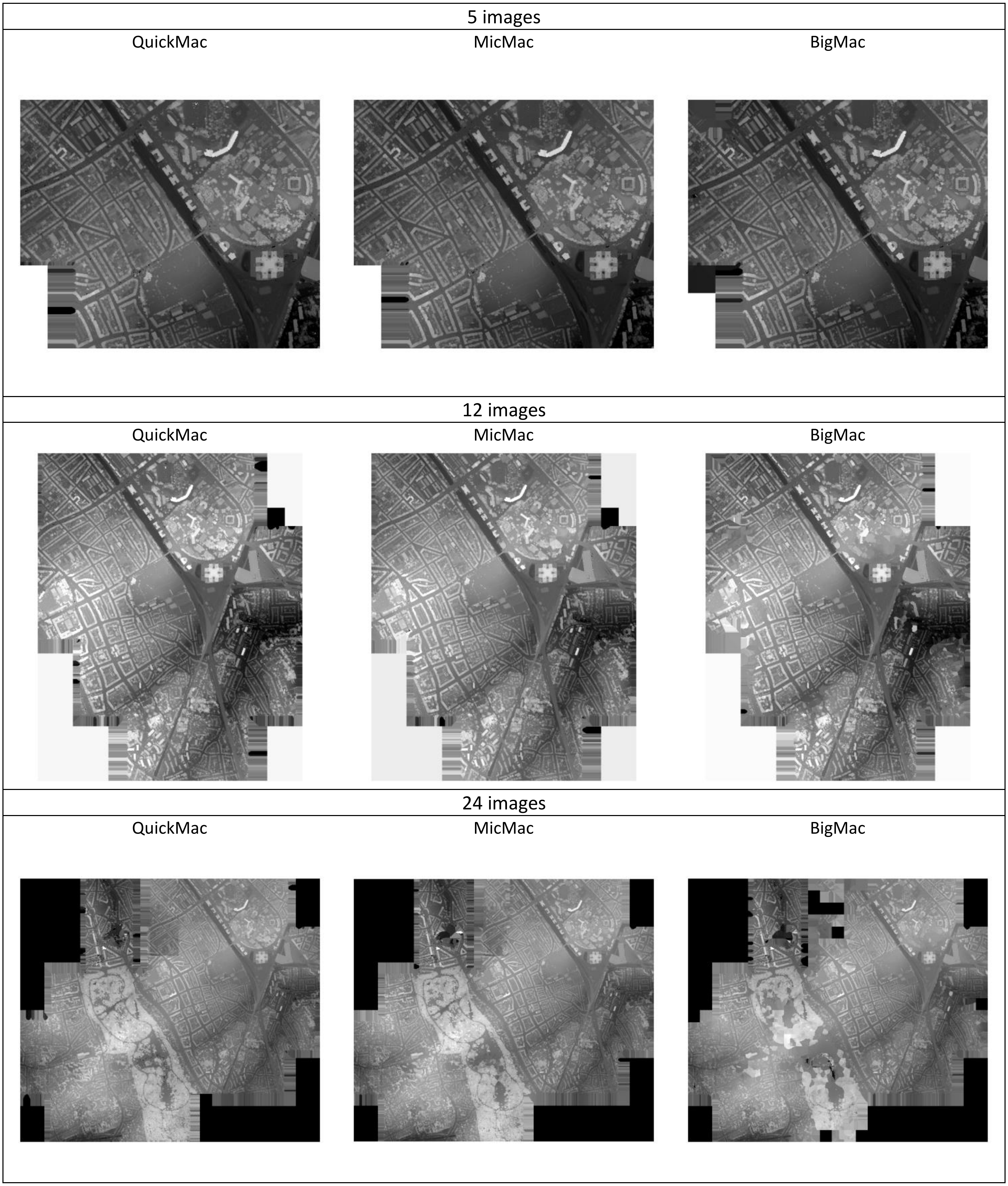


FIGURE 18: DSM (GLOBAL VIEW) FOR DIFFERENT C3DC MODES AND DIFFERENT NUMBERS OF IMAGES (FROM QGIS)

Just as for the dense point cloud, global views show that some areas (such as forest) are not modelled in high resolution (high resolution of `C3DC`, `BigMac` mode).
Relative heights are satisfying and similar, though well-modelled areas seem more precise in high resolution (more accurate geometries of objects). Detailed views show that a better resolution leads to more precise edges.

For comparison, the following images are a Lidar DSM[60] (pixel size: 1m) produced on the same area.

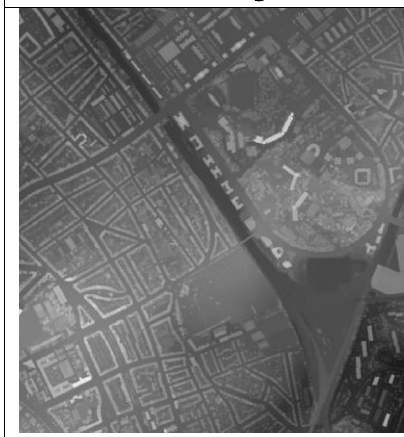
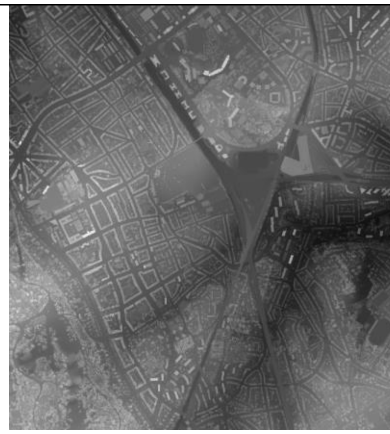
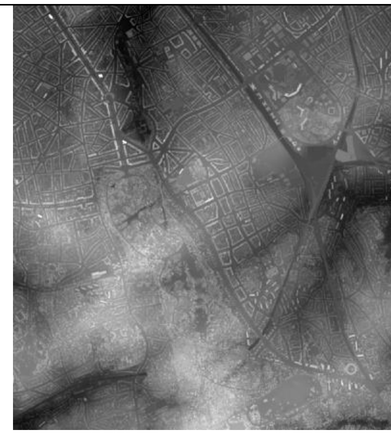
Area of the 5 images subset	Area of the 12 images subset	Area of the 24 images subset
		

Figure 19: DSM for reference (Lidar)

Please note that the **georeferencement does not depend on `C3DC` or `Pims2MNT`**, it is completed with `CenterBascule` (or `Campari` if used). Thus, `AperiCloud` (sparse point cloud), `C3DC` (dense point cloud), `Pims2MNT` (DSM) and the orthophotos all use the same georeferencement[61].

[60] Source: Agentschap voor Geografische Informatie Vlaanderen (agiv.be), 2014
[61] You can find some georeferencement error values for this particular dataset in the chapter *How to choose parameter values, II) Resolution for computation of tie points* about Tapioca, Table 4.

Frequent errors

The error messages in MicMac are rarely explicit. MicMac is still under development.

To track errors, please:

- Check the syntax. Beware in particular of regular expressions, the order of the parameters (type `-help` when in doubt), the orthography of the named parameters, and the name and path of every input file.
- **Check the residuals** of steps with adjustments or transformations (`Tapas`, `CenterBascule`, `Campari`…). Remember that you can save the information of the terminal into a text file, and that MicMac automatically generates XML files with the residuals.[62]
- Check which files were generated by the command that failed and inspect them. You can find the list of the generated files for each step in the tutorial. You can compare them with files of a dataset successfully modelled (if you have one).
- Please visit the MicMac forum[63] for help on any other error you may struggle with.

Here are some frequent errors you may face and what they refer to.

```
WARN  !! , for camera Vexcel Imaging GmbH UltraCam Eagle f210 cannot
determine focale equiv-35mm
add it in include/XML_User/DicoCamera.xml
```

The dimensions of the sensor are unknown. You should add them to *DicoCamera.xml*, or add the 35 mm equivalent focal length instead. Please refer to the chapter ***How to add the needed metadata, I) Add the focal length***.

```
|   Sorry, the following FATAL ERROR happened
|
|    cMetaDataPhoto::Foc
```

The focal length is unkown. You want to add it to the metadata of the images. Please refer to the chapter ***How to add the needed metadata, I) Add the focal length***.

```
  === NON INIT : 09531Y16.tif
------------------------------------------------------------
|   Sorry, the following FATAL ERROR happened
|
|    aBestCam==0
```

There is not enough or no tie points for one of the images (here, *09531Y16.tif*). Check that this image overlaps largely with the rest of the images. You may also try to run `Tapioca` again with a different mode or a different resolution. You can try and add new images (if you have some) to strengthen the weak overlapping area.

---

[62] Tapas, Campari: *Residus.xml* in the orientation directory. CenterBascule: *Result-Center-Bascule.xml* in the orientation directory.
[63] MicMac forum: http://forum-micmac.forumprod.com/

```
|   Sorry, the following FATAL ERROR happened
|
|    Not enough samples (Min 3) in cRansacBasculementRigide
```

You need more ground control points (GCPs) to be able to transform the coordinate system of the relative orientation into the absolute system. If you work with the centres (CenterBascule), you need at least 3 images. Indeed, you need at least 3 points known in both the initial and the final system to apply the transformation (it is mathematics).[64] If you already use 3 images or more, you may want to check that their centre is not null in the orientation XML files of the images, which are stored in the orientation directory used in the command line that led to this error (e.g. *Ori-GPS_Georef*).

The centre (position of the camera) of each image is stored in a file named *Orientation-[image_name].xml* in the orientation directory, where a <Centre> tag marks it. Here is an example of coordinates for centre in such a file:

```
<OrientationConique>
     …
     <Externe>
          <Time>0</Time>
          <KnownConv>eConvApero_DistM2C</KnownConv>
          <Centre>
          151302.369999999995 168062.410000000003 3111.19000000000005
          </Centre>
```

If you use a stereo pair, you need GCPs to be able to apply the transformation.

```
Sorry, the following FATAL ERROR happened
 Radial distortion abnormally high
```

The estimation of the coefficients of distortion failed. The least squares adjustment step did not fail but the values found are too high to be realistic, and the model cannot be built. You may want to start over the estimation of calibration but on a subset of images that may be more appropriate (get rid of images poorly orientated relatively just for the calibration)[65], or with another model of distortion (RadialExtended, Fraser…)[66]. You can also use more images if you have more.

You can run Tapas on a subset of images, and then use the calibration obtained on this subset for the whole dataset using the mode AutoCal (meaning that the calibration is based on the InCal parameter) and the parameter InCal:

```
mm3d Tapas RadialStd "img[0-7].tif" Out=Calib
mm3d Tapas AutoCal ".*tif" InCal=Calib Out=Relative
```

[64] See the MicMac documentation (pdf) to check out the mathematical equations for more details.

[65] The coefficients of calibration (values specific to each camera-lens pair and the settings for the acquisition) are estimated by least squares adjustment with distortion measurements taken from a set of images (taken with the same camera and unchanged settings). The adjustment algorithm computes the coefficients of the distortion model chosen by the user (RadialStd, Fraser…) based on the distortion measurements of the dataset. For the estimation to be optimal, the images must vary the angles of view, the distance from the photographed details, the orientation… Same details should be photographed from very different angles to have more meaningful measurements of distortion.

[66] See the chapter *How to choose parameter values, III) Choose a model of calibration.*

Similarly, with the parameter `InOri` you can compute a first relative orientation and apply its parameters to the rest of the dataset (there is a re-estimation). It enables the user to give more weight to the subset on which the first orientation is calculated. Please refer to the chapter ***Reuse a model of calibration*** for more information.

```
|   Sorry, the following FATAL ERROR happened
|
|    very singular matrix in Gausj
```

The least squares adjustment diverges. It is most probably directly due to the configuration of the photographs. You may need more tie points, larger overlapping… You can try to add some images to the dataset (if you can) to help the computation of the relative orientation by providing more information, choose another mode or a higher resolution for `Tapioca`, work on the colours of the photographs (if possible) to emphasise texture, force some parameters to probable values to help the least squares adjustment process[67]…

There is no error, the process terminates normally without crashing or warning but your point cloud is empty.

Check the residuals of Tapas.

In the terminal:

```
RES:[09447Y16.tif][C] ER2 -nan(ind) Nn 0 Of 9230 Mul 1415 Mul-NN 0 Time
0.162
RES:[09448Y16.tif][C] ER2 -nan(ind) Nn 0 Of 17044 Mul 2185 Mul-NN 0 Time
0.267
RES:[09449Y16.tif][C] ER2 -nan(ind) Nn 0 Of 10850 Mul 1408 Mul-NN 0 Time
0.167
| |  Residual = -nan(ind)
| |  Cond , Aver 35.7827 Max 44.9832 Prop>100 0
```

In the ***Residuals.xml*** file:

```
<OneIm>
…
   </OneIm>
   <OneIm>
      <Name>09449Y16.tif</Name>
      <Residual>-nan(ind)</Residual>
      <PercOk>0</PercOk>
      <NbPts>10850</NbPts>
      <NbPtsMul>1408</NbPtsMul>
   </OneIm>
   <AverageResidual>-nan(ind)</AverageResidual>
   <NumIter>3</NumIter>
   <NumEtape>3</NumEtape>
```

It reads 0% of points were kept for the computation. It is quite similar to the divergence error above (`very singular matrix in Gausj`): it is most likely due to the configuration of the photographs (lack of tie points, narrow overlapping…). Please read the advice about the `very singular matrix in Gausj` error.

---

[67] You can force the algorithm not to estimate some parameters. See the options in the documentation of Tapas (e.g. LibFoc=0 freezes the focal length).

It shows once more that **it is important to check the residuals**. The adjustment went wrong but did not crash, and the modelling failed without any raised error. Only the residuals show that there is an issue in the processing.

```
For Pat= [NKS-Set-OfFile@MMByPairFiles.xml]
------------------------------------------------------------
|   Sorry, the following FATAL ERROR happened
|
|    Empty pattern
```

This error may occur when you want to compute the point cloud with `C3DC`. The file `MMByPairFiles.xml` is the list of the images used for the dense correlation. It is in the main directory. It is supposed to be the list of the names of all the images of your dataset. If this error occurs, this file is probably empty. It means that the algorithm could not find the images that make up the model. This error is probably due to another error that occurred in a previous step. In particular, check that the percentage of points kept during the bundle adjustment (`Tapas`, `Campari`) is not 0%. Please refer to the previous error about that.

```
|   Sorry, the following FATAL ERROR happened
|
|    Arg0 is not a valid format specif
```

You just made a syntax error. Check the quotes, the orthography of the tools, modes and parameters, the order of the arguments…

```
For Key-Or-Pat=img01.JPG Dir= ./
------------------------------------------------------------
|   Sorry, the following FATAL ERROR happened
|
|    Empty list for StdGetListOfFile (one of the input file name is wrong)
```

This specific error occurred with `Tapas`, but you may have different error messages involving an **empty pattern**. Please check that the name of the images is correct. In particular, the images must be in the main directory (otherwise, you need the full pattern and not just the names of the images). Check also the extension (JPG, jpg, tif…), especially if you copied and pasted the commands. Beware that regular expressions are case-sensitive.