Notes on Creating Real Data for Testing and Validating 3D Reconstruction Methods from Agisoft Photoscan

EnCoV, IP, UMR 6602 CNRS, Université Clermont Auvergne / SIGMA, France

Corresponding author: Mathias Gallardo

`Mathias.Gallardo@gmail.com`

September 26, 2018

Most of methods developed in EnCoV deal with 3D reconstruction and require ground-truth 3D shape to test and validate them. We propose in this document one pipeline using the Structure-from-Motion (SfM) technique of PhotoScan Agisoft and the camera calibration of Agisoft Lens to reconstruct 3D textured shapes, *i.e.* a digital 3D model with its texture, from a set of RGB images.

Note that other methods such as structured-light system can be used to acquire ground-truth 3D shapes.

Contents

1   Overview

To test and validate Shape-from-Template (SfT) methods, you need the following data:

- the *object template*, the 3D shape of the object of interest in a rest pose

- the *parameters of camera* used to construct the object template

- the *deformed shapes*, the 3D shape in each view of the object in deformation. These deformed shapes are the 3D shapes that we want to reconstruct.

- the *parameters of the camera* used to construct the deformed shapes

Definitions   By *rigid view* we refer to an image where the object is not deformed. By *deformed view* we refer to an image where the object is deformed.

In this context, a 3D shape of an object is defined by three groups of files:

- a 3D texture-mapped model of the object of interest, contained in 3 files: `.obj`, `.mtl` and `.png` (or `.tif`). The `.obj` file contains the 3D information, the `.mtl` file contains appearance data and the `.png` file is the texture map and has to be indicated in the `.mtl` file.

- the intrinsic camera parameters: `camera.xml`.

- the extrinsic camera parameters: `PSCameras.xml`. This file contains all the poses (rotation and translation) of the different rigid views used to reconstruct the shape of the object.

This object of interest is the *object template* or the *deformed shapes*. Note that, if the same camera is used to acquire the images to build the *object template* and the *deformed shapes*, the intrinsic camera parameters will be the same. Here an example: if we want to test a SfT method in an object on 5 input images showing 5 object deformations, we will need one ground-truth 3D shape for the template and one for each input image.

Layout   To construct a 3D shape of an object, we need first calibrate the camera (*i.e.* estimate all their intrinsic parameters) and then reconstruct in 3D the shape of an object. To do that, we use two Agisoft softwares since they are easy to use and reasonably fast to compute.

2 Camera Calibration: Agisoft Lens

2.1 Which versions?

We recommend to choose the following version: v 0.4.1 (1718 build). The reason is that the format of the data was changed in the next versions. For instance, in the newer versions, the principal point is defined as a offset from the center of the image, while, in the one we recommend and in the older ones, the principal point is defined from the top-left corner. We found that with this version the calibration results are similar to the one of the `cameraCalibrator` of Matlab.

Note that if however you decided to use a newer version of Agisoft Lens, be careful with the principal point format when you will use the data in another environment, such as Matlab.

2.2 Data format

Note that the camera parameters in Agisoft Lens are defined differently from Bouguet's Toolbox [3] or `cameraCalibrator` [5] of Matlab. Note that the latter uses the former: their data format are similar. Table 1 gives some relationships between Agisoft Lens v0.4.1 (1718 build) and [3]. We can find all the details at Bouguet's Toolbox [1] and [3].

	Agisoft Lens v0.4.1 (1718 build)	[3] and Camera Calibrator [5]
Radial distortions	k1	$k_c(1)$
	k2	$k_c(2)$
	k3	$k_c(5)$
Tangetial distortions	p1	$k_c(3)$
	p2	$k_c(4)$
Skew coefficient	skew	$f_x \times \alpha_c$

Table 1: Equivalence between the camera calibration formats of Agisoft Lens v0.4.1 v0.4.1 (1718 build) and [3].

2.3 Inputs

- a set of images where the calibration pattern, shown in the figure 1, is visible in at least 10 different viewpoints. Figure 2 shows an example of a set of calibration images. §2.7 describes how to take such set of images.

2.4 Outputs

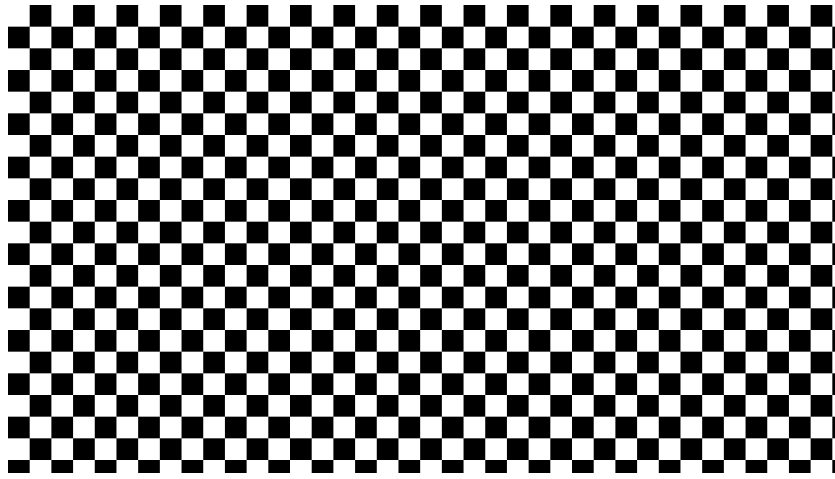- a .xml file that we usually name `camera.xml`

Figure 1: The calibration pattern.

2.5 How to Capture Good Images for the Camera Calibration?

1. in the software, click in the calibration icon to display the pattern. Do a screenshot and print it.

2. be sure that the printed pattern is not too glossy because specularities make the calibration harder

3. be sure that the calibration pattern is glue over a flat surface and that there is no air bubbles left under the sheet of paper

4. selecting (manually or automatically) the focal length of your camera by taking a sharp image of your object of interest: the object has to be fully visible in this image and not to close to the image edges

5. fix the focal length of your camera by switching the camera to manual (see the physical button of the camera optic)

6. note the distance between the camera and the object: we called it $d_{o \to c}$

7. take at least 10 different viewpoints of the calibration pattern: check that the images are sharp, that there is no strong specularity in the big part of pattern and that the distance between the pattern and the camera is close to $d_{o \to c}$. You can take many photos as you want

2.6 General Procedure

1. open Agisoft Lens

2. click on Tools → Add Photos and select the calibration images
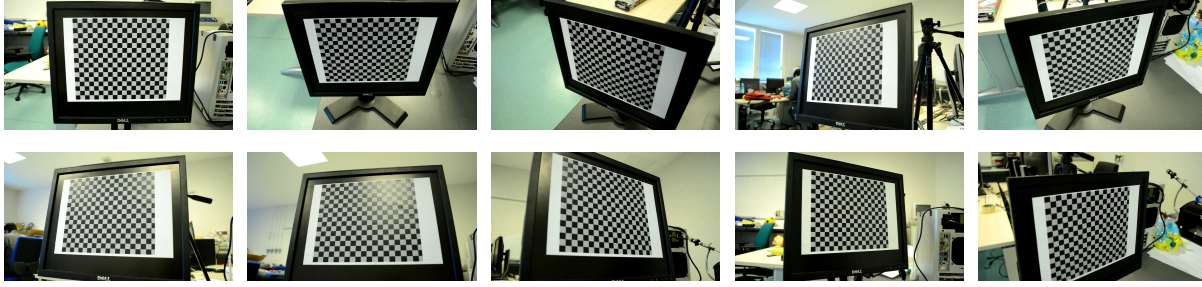
3. click on Tools → Calibrate

Figure 2: Example of calibration images required to perform the camera calibration.

4. a window such as the figure 3 will appear: select the same calibration parameters

   - the center principal point: cx, cy

   - the radial distorsion parameters: k1, k2, k3

   - the tangential distorsion parameters: p1, p2

   Note that, with current commercial and industrial cameras, the skew can be assumed to equal to zero and does not need to be estimated.

5. click on Ok: the camera calibration is now being perfomed

6. figure 4 shows the result of camera calibration

7. check if the camera parameters are coherent (see §2.7)

8. if they are coherent, click on File → Save calibration

2.7   How to Check if the Camera Parameters are Coherent?

- the Image width and height have to be the same as the calibration images

- the center principal point, x and y, has to be close to the center of the image.

  If the value of the principal point is closer to 0 than half the image size, then you are using a newer version of Agisoft Lens than the one recommended. Good value of x- and y-components are respectively at $width./2 \pm width \times 0.1$ and $height./2 \pm height \times 0.1$.

- the skew has to be equal to zero since we uncheck "Fit skew"

- k1, k2 and k3 are the radial distortion parameters.

  If the original images do not look to be distorted, k1 has to be less than 0.1 in absolute value.

If k2 is superior than 100 in absolute value, verify that the original images are clearly distorted and, if it is not, that means the calibration images do not present enough different viewpoints.

k3 can be superior than 100 in absolute value. The sign of the radial distortion can be also checked as figure 5 indicates.

- p1 and p2 are the tangential distortion parameters. For the cameras of the lab, p1 and p2 has to smaller than 0.1.

- another verification can be done by undistorting the images (see §3.7).

- do not forget to click on File → Save calibration to save the camera parameters.
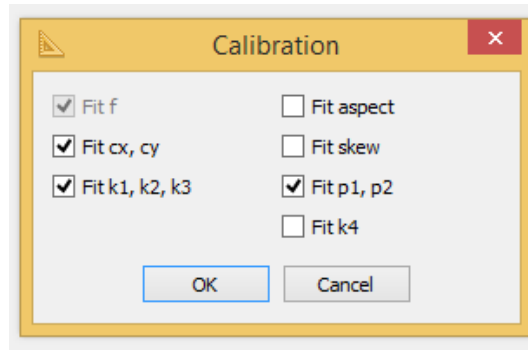

Figure 3: Calibration parameters to estimate in Agisoft Lens.

3   3D Reconstruction Using SfM: Agisoft PhotoScan

3.1   Which versions?

We recommend to choose the following version: v 1.2.6 (2331 build). The reason is that the data format of the camera calibration from Agisoft Lens is readable by this version. Older versions are also good.

3.2   Inputs

- a set of images of the object of interest in a rigid pose all over the sequence. The number of images can vary depending on the complexity of the surface and its texture. See §3.4 for tips to acquire pictures which lead to good 3D reconstructions.

- the camera calibration file `camera.xml` of the camera which is used to take the set of object images.
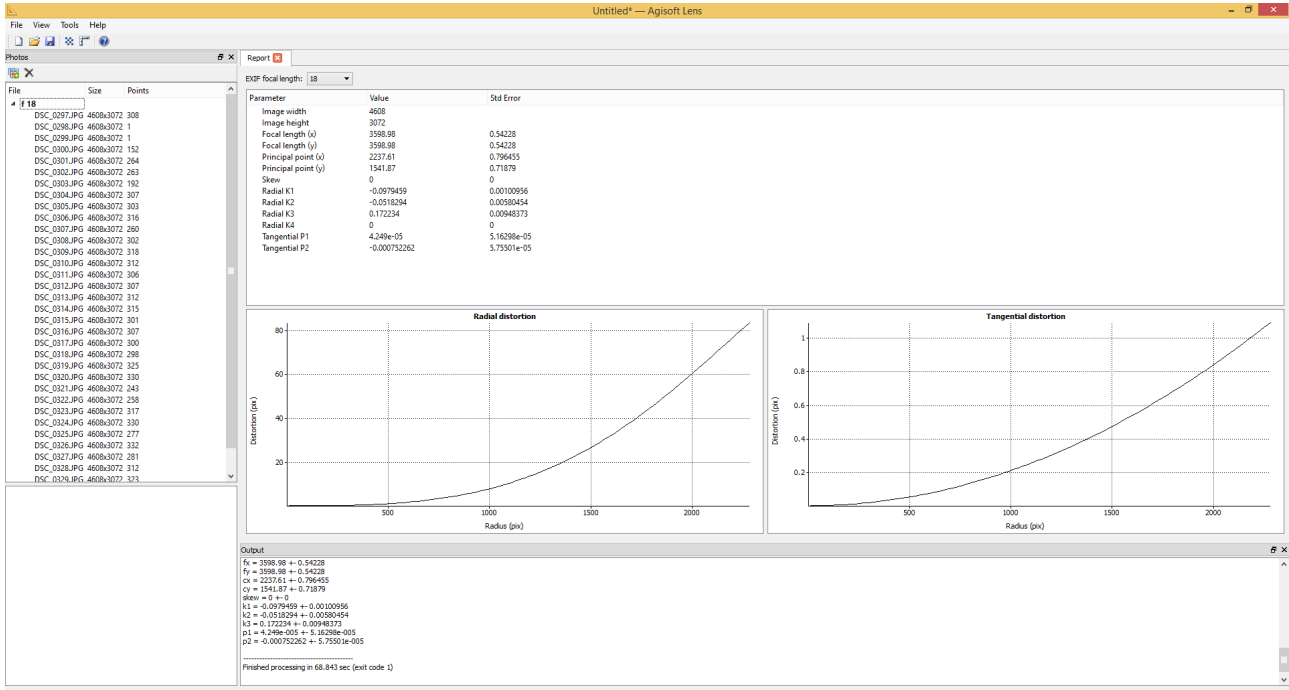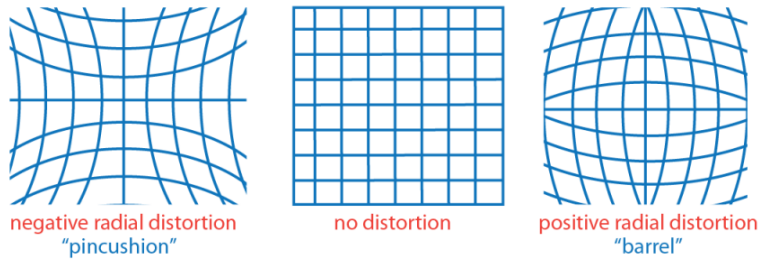
Figure 4: Results of camera calibration.

Figure 5: Relationship between the sign of radial distorsion parameters and its visual effect. Source: [4].

3.3   Outputs

- a 3D texture-mapped model of the object of interest contained in 3 files `.obj`, `.mtl` and `.png` (or `.tif`)

- the extrinsic camera parameters that we name `PSCameras.xml`. It contains the rigid transformation (rotation and translation) between the world coordinates frame and each camera coordinates frame, which corresponds to each object image.

- the masks of each image to extract the foreground from the background

- the undistorted images used to do the 3D reconstruction

3.4   How to Capture Good Images for the 3D Reconstruction?

There are already good manuals on internet. In the folder `./docs` attached to this report, you will find [2] and another document with some similar informations. The *Chapter 2. Capturing photos* of [2] provides useful information. Here we provide some information that we found relevant for our works. There is no theoretical conditions. Therefore, one has to train his intuition to know a priori when an object will be correctly reconstructed.

3.4.1   Problem 1: When the Object Surface is Well-Textured, but When the Texture is Repetitive and the Shape is Symmetric

Problem   This scenario can make the reconstruction impossible or wrong. The figure 6 shows two examples of ambiguous objects and shapes. Both sides are not enough different, but each of them is sufficiently well-textured to obtain a 3D reconstruction of one side. Therefore, it is possible to reconstruct each side separately, but, if we align the pictures of both sides, we will see that the solution proposed is just one face. This is due to the fact that the texture and the shape of both sides are not enough different.

Figure 6: Examples of ambiguous object shapes. Both sides are quite similar and the shape is not enough discriminative.

Solution   Figure 7 illustrates some solutions to this issue. These new sequences lead to good reconstructions.

- make the texture very different between all the sides

- place the object (if the object is deformable) such that the shape is not too symmetric

3.4.2   Problem 2: When the Object Surface is Poorly-Textured

Problem   In this scenario, the object is not enough textured. This makes the reconstruction very difficult even impossible.
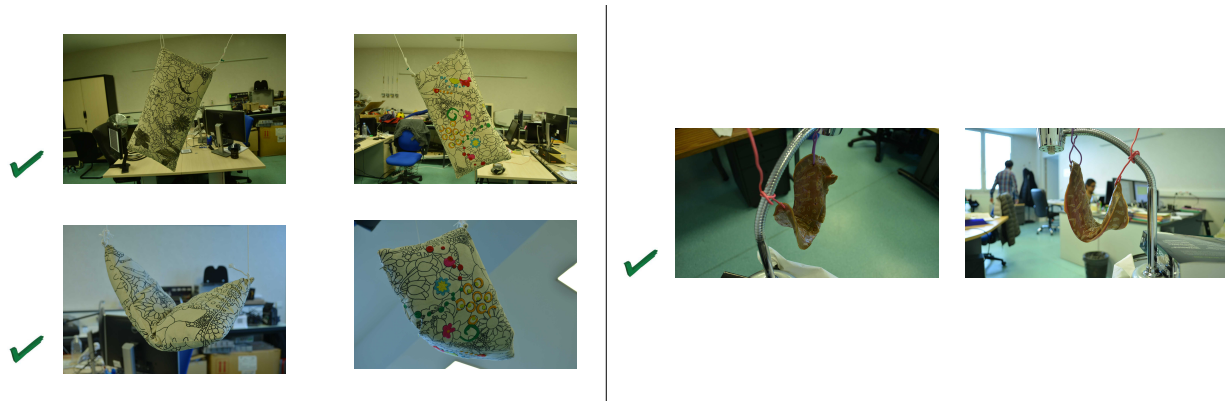
Figure 7: Examples of good shapes that lead to a good reconstruction.

Solution

- add texture all over the surface

- do not use masks and capture the pictures with a textured background. In deed, a textured background
  can be very beneficial for this case since it provides more constraints for the reconstruction. However,
  be sure that the background does not change during the sequence.

3.5   Procedure

A good tutorial to perform 3D reconstruction with Agisoft PhotoScan is provided in this following link:
`http://www.agisoft.com/pdf/PS_1.1%20-Tutorial%20(IL)%20-%203D-model.pdf`.

3.6   Good practices

- save the project as soon as possible

- save as soon as possible the masks in a folder named `./masks` because segmenting the images takes
  time and has to be done again if the software crashes

3.7   How to Undistort Images from Agisoft PhotoScan?

1. open Agisoft Lens

2. click on Workflow → Add Photos and select the images to undistort

3. click on Tools → Camera Calibration

4. load you camera calibration file `camera.xml` and check "Fix Calibration" (so Agisoft PhotoScan cannot
   change the camera parameters)

5. click on Tools → Export → Undistort Photos: figure 8 will appear

6. be sure that the box of "Center principal" point is unchecked such as the figure 8 shows

7. click on "OK" and select a folder to store the undistorted images
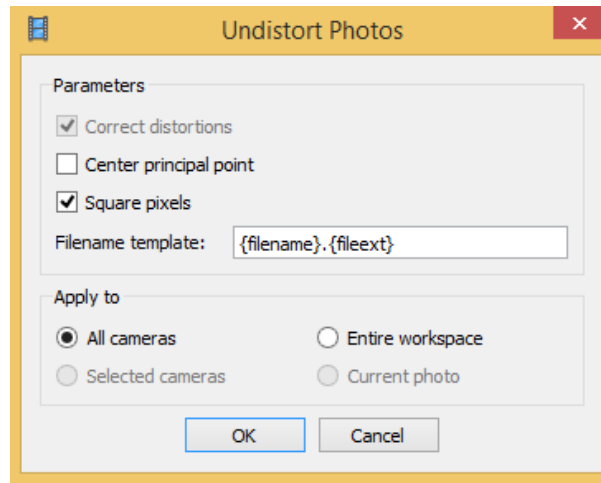

Figure 8: The options window to undistort the images in Agisoft PhotoScan.

References

[1] Agisoft Lens. Agisoft Lens User Manual - version 0.4.0. `http://downloads.agisoft.ru/lens/doc/en/lens.pdf`.

[2] Agisoft PhotoScan 1.2. Agisoft PhotoScan User Manual, Standard Edition, Version 1.2. `http://www.agisoft.com/pdf/photoscan_1_2_en.pdf`, 2014.

[3] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. `https://www.vision.caltech.edu/bouguetj/calib_doc/`.

[4] MathWorks. cameraparameters class. `https://ch.mathworks.com/help/vision/ref/cameraparameters-class.html`.

[5] Matlab. Camera calibrator. `https://ch.mathworks.com/help/vision/ref/cameracalibrator-app.html`.