

Notes on the Use of Multiple Image Sizes at OpenCV stereo

Antonio Albiol

April 5, 2012

Abstract

This paper explains how to use different image sizes at different tasks done when using OpenCV for stereo processing.

1 Introduction

Stereo Processing is the set of tasks that must be done in order to estimate depth of points in a scene from the images coming from two cameras placed side by side. Stereo Processing comprises the following steps:

1. Calibration. This step consists in acquiring a certain number of image pairs containing a chessboard (or other pattern). This process is normally performed off-line before the stereo pair is actually used for any application. For better results, high resolution images should be used if possible.
2. Image Acquisition. Stereo processing is normally used to estimate depth of points in a given scene. Quite often real time operation is required. This imposes limitations on the size of the input images.
3. Depth estimation. Before depth is estimated input images must be rectified to remove distortion and to align them properly. A pair of left and right rectified images is obtained and then, disparity is estimated from this pair of images. In order to balance between computational load and quality of results it is often interesting to be able to rectify input images to a different size.

2 Why different sizes

Very often, one camera can provide images at different resolutions. Also, frame grabbers allow to select the size of image captures.

Different needs require different sizes. For instance, calibration requires the maximum accuracy. This is accomplished with images with the highest possible resolution. During calibration, time is not a major concern, and normally it is not problem to use several seconds with one pair of images.

However real time applications require a clever choice of image size in order to balance between speed and accuracy.

3 Calibration Images

In this section we are going to give some hints about how to choose images for calibration.

3.1 Pixel Aspect Ratio

At least from my experience, I have seen that pixel aspect ratio of images used for calibration should close to one. This is important specially when you work with *interlaced video*. Normally only one of every second line is retained yielding an image that has a pixel aspect ratio (PAR) of 2. Before proceeding to calibration, this images must be vertically interpolated or horizontally decimated, otherwise calibration functions do not work properly.

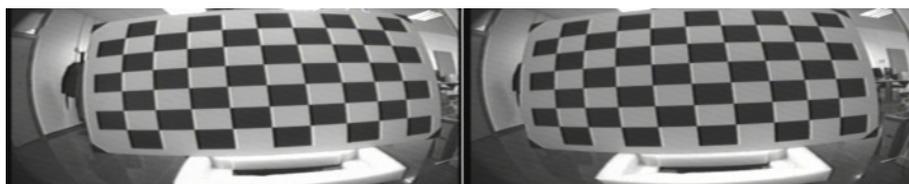


Figure 1: Example of Left-Right input pair having pixel aspect ratio $PAR = 2$.

3.2 Image Size

The largest possible image size should be used for image calibration even if we know in advance that this size will not be feasible for normal (real-time) operation.

If we have captured images with $PAR = 2$, it is better to interpolate in the vertical direction than to reduce the size in the horizontal direction. I have achieved good enough results using linear interpolation, but other interpolation approaches such as cubic are also possible. Interlaced video is a common cause of differences between input images and calibration images sizes.

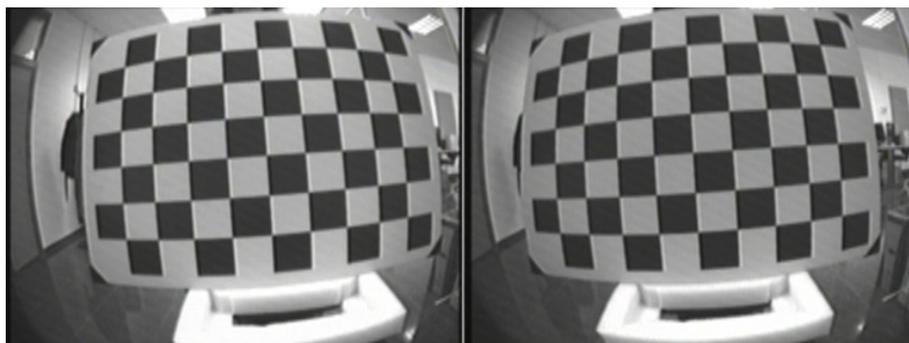


Figure 2: Example of Left-Right pair that has been vertically interpolated to get $PAR = 1$.

3.3 Field of view

In order to get good calibration results, the content of the calibration images is more important than the number of images. For better calibration I would recommend to *use a chessboard as large as possible* and take images with the following characteristics:

- Try to place the chessboard as close to the cameras as possible so that:
 - Most image area is covered by the board in each pair of images (See sample images in this document)
 - The board forms an angle as large as possible with the image plane. The worst views are those in which the chessboard is parallel to the image plane.
- In the case of wide baselines, it is better to estimate first the cameras intrinsics using 3 sets of images:
 - A set of images for calibrating left camera intrinsics, with the chessboard as close to the left camera as possible.
 - The same but for the right camera.
 - A set of images, in which the board is seen *simultaneously* by both cameras to estimate stereo extrinsics with the flag `CV_CALIB_FIX_INTRINSIC`.
- In our experience, images where the chessboard is far from the cameras (compared to the chessboard size) are of little value, since weak perspective conditions are met in this case what makes almost impossible to estimate well camera intrinsics.

3.4 Calibration Results

The results of stereo calibration are:

- Two matrixes of intrinsic parameters, M_L and M_R (in OpenCV left camera is normally camera 1 and right camera is camera 2). Left and right cameras are understood in the same sense as left and right eyes. This matrixes correspond to a calibration image size. This is the reason why the image size, w_{cal} , and h_{cal} (in pixels), from which M_L and M_R were obtained must be kept together with the matrixes. Intrinsic matrixes have the following internal structure:

$$M_L = \begin{pmatrix} f_{Lx} & 0 & c_{Lx} \\ 0 & f_{Ly} & c_{Ly} \\ 0 & 0 & 1 \end{pmatrix} \quad M_R = \begin{pmatrix} f_{Rx} & 0 & c_{Rx} \\ 0 & f_{Ry} & c_{Ry} \\ 0 & 0 & 1 \end{pmatrix}$$

- Radial Distortion Parameters D_L , and D_R of each camera. These parameters **do not depend of the image size or PAR**.
- Extrinsic Rotation Matrix R : This matrix expresses the relative rotation between both cameras and so it **does not depend of the image size**.
- Translation Vector T : This vector contains the translation (in mm) that exists to move from the right camera location to the left one; so it **does not depend of the image size**.

4 Input Images

Input images are those for which we want to compute stereo. If these images happen to have a different size of those used for calibration, there are two options:

- Resize input images so that they have the same size as calibration images, and use calibration data exactly as obtained from the calibration.
- Keep input images in their original size, and adapt the calibration values to the new image size.

First option normally requires some CPU time to rescale the images, and so it is more computationally demanding. Moreover, if calibration images were very large it may happen that we are upscaling images by a large factor unnecessarily.

We normally prefer the second option.

Let w_{cal} and h_{cal} be respectively the width and height of the images used for calibration that yielded M_L , M_R , D_L , D_R , R and T .

Let w_i and h_i be respectively the width and height of the input images. We try to compute M'_L , M'_R , for the input images size. The other calibration data, namely D_L , D_R , R and T remain the same in the case of a different image size.

First we compute horizontal and vertical scale factors:

$$s_x = \frac{w_i}{w_{cal}} \quad s_y = \frac{h_i}{h_{cal}}$$

$$M'_L = \begin{pmatrix} f_{Lx} s_x & 0 & c_{Lx} s_x \\ 0 & f_{Ly} s_y & c_{Ly} s_y \\ 0 & 0 & 1 \end{pmatrix} \quad M'_R = \begin{pmatrix} f_{Rx} s_x & 0 & c_{Rx} s_x \\ 0 & f_{Ry} s_y & c_{Ry} s_y \\ 0 & 0 & 1 \end{pmatrix}$$

```
Size inputImageSize;
inputImageSize.w = w_i;
inputImageSize.h = h_i;
cv::stereoRectify( M'_L, D_L, M'_R, D_R, inputImageSize, R, T,...
```

We have successfully used the above method in cases where calibration was done with $PAR \approx 1$ with input images with $PAR \approx 2$.

5 Rectified Images

Rectified images are those used to estimate disparity. Rectification parameters are computed with the function `cv::stereoRectify`. This function admits an input argument which is `Size newImageSize`, that allows the user to specify the size of rectified images. Most OpenCV example programs use the input image size for the rectified images. In our opinion this poses two drawbacks:

- Less flexibility. Being able to tune the rectified images size allows to control the computational load. Remember that the main computational cost is disparity computation, and that **halving the rectified images size reduces by 8 the computational load** (4 times less pixels and half disparity range).
- Maximization of the usable portion of image. Rectified images have exactly $PAR = 1$. That means that if input images have $PAR \neq 1$ images are cropped and scaled so that the rectified images have exactly `newImageSize` with square pixels. This normally implies discarding a portion of the original image. This effect is specially important if input images have $PAR \approx 2$!!

- In the case of severe radial distortion, since images are normally wider than higher, distortion is more important in the horizontal than in the vertical direction. Assuming that the rectified images have the same size as the input images, produces an unnecessary cropping in one dimension. Figure 3 illustrates this excessive cropping in the horizontal direction.

The function `undistortPoints` can be used to estimate the right proportions of the rectified images to maximize the useful area.



Input images with radial distortion and $PAR \approx 1$



Rectified images with `newSize=inputSize`



Rectified images with optimum choice for `newSize`

Figure 3: Importance of the good choice of rectified images size. Notice how the right choice for the rectified image size has kept a larger portion of the original content.