Detection of concentric circular patterns through filters, oval detection and metaheuristics.

Antoni Mauricio
Research and Innovation Center
in Computer Science
Universidad Católica San Pablo
manasses.mauricio@ucsp.edu.pe

Jorge López
Research and Innovation Center
in Computer Science
Universidad Católica San Pablo
jorge.lopez.caceres@usp.pe

***Abstract*—Camera calibration is one of the primary processes in computer vision, its correct calibration significantly defines efficiencies in more complex methods such as augmented reality, 3D reconstruction or application as SLAM (all cases require to obtain 3D spatial information). The calibration process is necessary to obtain 3D information from 2D images. There are different techniques based on photogrammetry and self-calibration. As a result, the intrinsic and extrinsic parameters of the camera are obtained. Much work has been done in the calibration and also in data pre-pos-processing (metaheuristics). Most authors work over methods based on two-dimensional template as the easiest path to perform and obtain the best results. However, the improvements realized with different metaheuristics can contribute to calibration process, even if these are normally not considered in calibration pipeline.**

Many metaheuristics which improve calibration process are presented and evaluated in Zhang [1]. These techniques optimally determine the calculation processes, eliminate noise in points coordinates and perform a non-linear search appropriate in a set of camera parameters. This paper aims to define the complete procedure to calibrate a camera using a flat template of concentric circular patterns and achieve optimal results with the process. By other hand, calibration process (and also segmentation for our case) can be improve in probabilistic model introduction in order to define a robust algorithm to detect the flat template in video with different sources of noise.

Keywords: Camera calibration, filters, metaheuristics.

1. Introduction

The calibration accuracy determine the accuracy of the measures that are carried out from the images. It is for this reason that it is essential to perform the camera calibration with full guarantees that the parameters obtained are like the real ones. This commitment implies both: the right choice of calibration method as well as the correct use of it. So, the calibration process should start by making a exhaustive review of the state of art over different calibration methods to choose the one that could get better results under defined conditions. Due to the large amount of work done in calibration field, it is an arduous and uncomplicated task choice of method and conditions for develop it.

The first step in calibration process is achieve to identify the flat template of concentric circular patterns, to do so, a segmentation algorithm is needed, most known segmentation algorithms and techniques are described in [2] and [3]. Segmentation process use features in common inside images to define an object, which even in human vision and perception also causes many confusions.

The second step, according to [4] is object tracking, this process allow us to follow an object of interest over a video, instead of segmenting per frame the whole video, which has highly-cost. The methods that have presented better performances in tracking are based on Monte Carlo and probabilities, considering that tracking could be described as Markov Chain [5]. We opted for Particle Filter over Extended Kalman Filter, which is the other most used method, not just due points described before but also for cost-benefit between the implementation, efficiency of the method and its computational cost.

The third step, is solve calibration equation, this approach runs from images distortions introduced by pinhole cameras (very common). Two major distortions are radial distortion and tangential distortion. Due to radial distortion, straight lines will appear curved. Its effect is more as we move away from the center of image.

The fourth step, imply to obtain fronto parallel projection using calibration camera parameters, obteined from step third. To do so, first we have to project corrected and undistorted images to our full windows in order to calculate collinearity (if points supposedly corrected look actually as original/ printed pattern), then using others parameters like homography matrix and perspective parameters as stop-criteria we can repeat calibration process until a convergence.

Finally, to finish our ideal pipeline, first a good flat template tracking has done, many position estimations should be performed (in better way) and finally a propper calibration process have to be performed, this part gonna be explored in future presentations.

2. Methods

This section presents the mathematical and theoretical definitions of points described before.

2.1. Detection Algorithm

The algorithm works each frame in gray scales. First we use median filter to reduce impurities. Then an elliptical kernell is used for dilatation process in order to remark black contours (or white contours expanding) to highlight the rings. We calculate an adaptive threshold to turn the image into black-white scale (binarization) without affecting the brightness. we invert the result and dilated them because it must give us a very reduced result. Median-term result is used as a mask to filter the noise in edges when applying canny filter to the original image with a Gaussian filter. to this result we scan for elliptic edges and keep them limited by size. From candidate objects we choose the ones that are their centers are concentric or close, the form to look for them is by means in a histogram (every center is stored). We only choose those that are concentric with at least one more. Pipeline as is presented in figure 1, runs as follow:

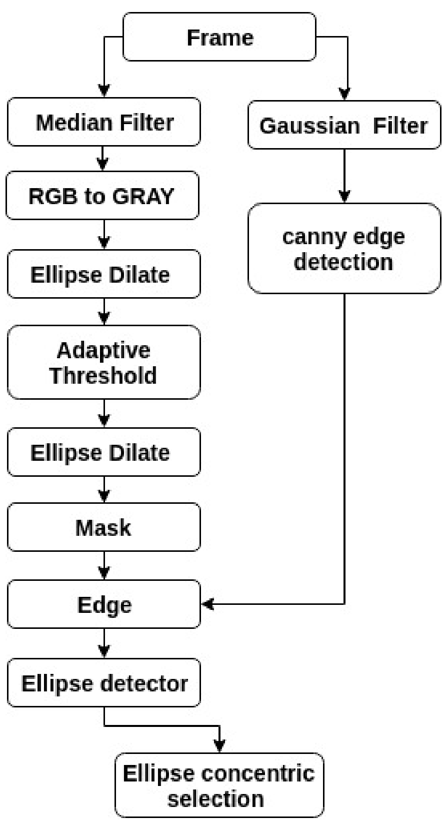

Figure 1. General pipeline

2.2. Tracking Algorithm

We are implementing a particle filter (PF) version, using the basic idea of image odometry (obteined from previous frames), doing so, we could get a naive pose estimation, but good enough to implement any error correction method.

The objective of a particle filter is to estimate the posterior density of the state variables given the observation variables. The particle filter is designed for a Hidden Markov Model (HMM), where the system consists of hidden and observable variables. The observable variables (observation process) are related to the hidden variables (state-process) by some functional form that is known. Similarly the dynamical system describing the evolution of the state variables is also known probabilistically.

For our work we are implementing a generic particle filter, which estimates the posterior distribution of the hidden states (pose estimation - X_k) using the observation measurement process (object detection - z_k) and the discretization of the movement respect to the previous state. Considering a state-space shown in figure 2.

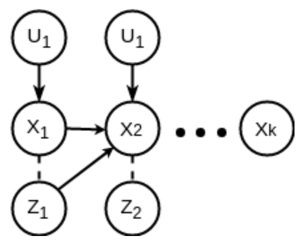

Figure 2. Markov chain used for particle filter

The pseudocode is presented below:

Algorithm Particle Filter (X_{t-1}, u_t, z_t):
$\hat{X}_t = X_t = \emptyset$
for $M = 1$ **to** M **do**
  sample $x_t^{[m]} \sim (x_t)$
  $w_t^{[m]} = \frac{p(x_t^{[m]})}{\pi(x_t^{[m]})}$
  $\hat{X}_t = \hat{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
end for
for $M = 1$ **to** M **do**
  draw i with probability $w_t^{[i]}$
  $x_t^{[i]}$ to X_t
end for
return X_t

.

When the pattern is lost, the algorithm tries to estimate the elements positions of the pattern using the previous frame completely detected. To do so, a cost function is evaluated between the possible elements positions of the previous state (x_{k-1}) with the elements of the new state (x_k) using a MSE (mean squar error) of possible poses.

2.3. Frame Distribution

For the correct calibration of the camera, it is necessary to have a quasi-equitable distribution of the patterns over the total of the scene, for which a density-based pattern distribution algorithm is applied, that is, the frames are selected for calibration based on how his spatial arrangement impacts on the average density and by sectors in the scene. The sector density method include quad segmentation over scene, in our case we define $4x5$ sectors due the window size and proportion. To set the maximum number of points per zone or density (p), we establish a maximun number of frames to use in calibration process (N_{frames}), then based on number of elements per arrange ($N_{arrange}$) and total number of sections ($N_{sections}$), we define equation 1:

$$p = \frac{N_{frames} * N_{arrange}}{N_{sections}} \tag{1}$$

The following algorithm presents the pseudocode of the density selection algorithm.

Algorithm Density Distribution (Vector of IFrames):
$p = N_{frames} * N_{arrange} / N_{sections}$
for $S = 1$ **to** $N_{sections}$ **do**
  for $F = 1$ **to** N_{frames} **do**
    Evaluate density with Iframes[F]
    if Pass evaluation **then**
      Add frame to OFrames
    end if
  end for
end for
return Vector of OFrames

2.4. Camera Calibration

According to [6], camera distortion is solved using five camera parameters, known as distortion coefficients:

$$Distorsion - Parameters = \{k_1, k_2, p_1, p_2, k_3\}$$

Where $K_n = n^{th}$ are the radial distortion coefficients and $P_n = n^{th}$, the tangential distortion coefficients.

The radial distortion is irregular, the most commonly encountered distortions are radially symmetric, or approximately so, arising from the symmetry of a photographic lens.

$$\begin{bmatrix} X_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{bmatrix} \tag{2}$$

Similarly, another distortion is the tangential distortion which occurs because image taking lense is not aligned perfectly parallel to the imaging plane. So some areas in image may look nearer than expected. It is solved in 3:

$$\begin{bmatrix} X_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \end{bmatrix} \tag{3}$$

In addition to this, we need to find a few more information, like intrinsic and extrinsic parameters of the camera. Intrinsic parameters are specific to a camera. It includes information like focal length (f_x, f_y), optical centers (c_x, c_y) and others. It is also called camera matrix. It depends on the camera only, so once calculated, it can be stored for future purposes. Extrinsic parameters corresponds to rotation and translation vectors which translates a coordinates of a 3D point to a coordinate system. It is expressed as a $3x3$ matrix:

$$Camera - Matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

In order to reduce camera calibration process we decided to use **calibrateCamera** function from *OpenCV*. **calibrateCamera** function uses ordered arrays obteined from detection/tracking algorithm and windows size, and returns camera calibration matrix, distortion coefficients $(k_1, k_2, p_1, p_2[, k_3[, k_4, k_5, k_6]])$ of 8 elements, rotation and translation vectors.

2.5. Fronto Parallel projection

3. Results

This section presents results from all algorithms an modifications, errors included. So, each subsection shows results step by steps focusing on camera calibration and fronto-parallel parameters.

3.1. Pattern Recognition

Based on Gaussian Filter and Canny Edge Detection, we perform new results for feature recognition over elliptical patterns, those results are presented on figure **??**. As can be seen, the added features turn our pipeline in a quite robust system, with less deficiencies for situations that in the previous samples were adverse.

3.2. Tracking

For tracking, we use the first two frames in order to estimate initial conditions (needed for future estimations), this first initial variables are: angle and descriptors pose (patters points obtained from previous steps). Figure 3 shows average results from tracking.

The main goal was solved, the method understand the pattern as a full element, which is composed by points (elements centroids). The bug is caused by mal-detection of elements in rotation (correlation cost function could fail in this case).

As can be seen in figure 3, even in rotation the pattern are tracked. In some frames, a partial or total mal-detection happens (in middle of frames showed) then the cost function machting starts to work.

In case of partial lose or total lose, the algorithm try to match elements from previous frame detected and new one. This is show in figure 4.

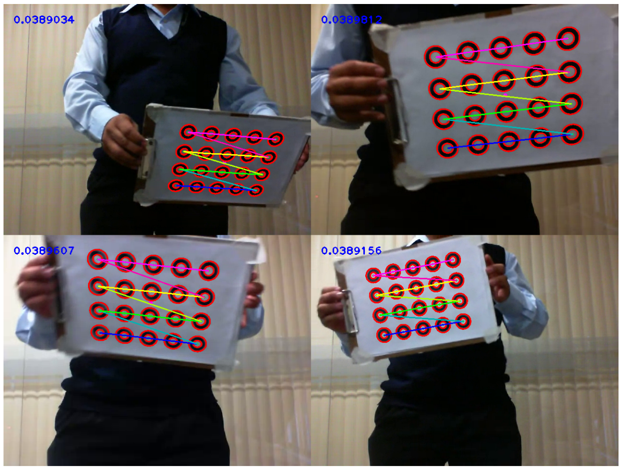

Figure 3. Concentric rings tracking results, the frames shows the results of Particle Filter Tracking in video, the video is presented by frames from top to down

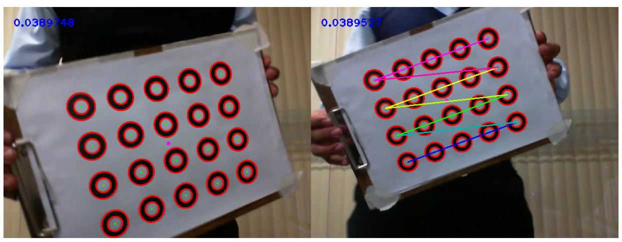

Figure 4. Matching function working over a pattern with partial or total mal-detection.

3.3. Calibration

After tracking, and using OpenCV calibration function for 3 patters (P): Concentric rings (2), Assimetric disks (1) and Chessboard (0), to calibrate 2 cameras: PS3 and LifeCam. (c_x, c_y) is a principal point that is usually at the image center (f_x, f_y) are the focal lengths expressed in pixel units

The Table 1 shows full results considering number of frames (N_F) used for calibration in LifeCam camera, while 2 shows results in PS3 camera:

P	N_F	F_x	F_y	C_x	C_y	**RMS**
0	51	588.038	589.47	315.49	228.460	0.7608
1	25	622.340	625.00	340.08	205.049	0.1734
2	25	615.420	617.19	325.00	223.590	0.1817

TABLE 1. LIFECAM CALIBRATION RESULTS

P	N_F	F_x	F_y	C_x	C_y	**RMS**
0	35	854.051	856.016	348.220	250.09	0.2559
1	35	845.390	846.885	338.313	248.68	0.2371
2	27	844.195	847.380	340.606	260.84	0.2081

TABLE 2. PS3 CALIBRATION RESULTS

Corrected frames using calibration parameters are shown in figure 5, 6 and 7 for concentric rings, chessboard and assimetric disks patterns, for this purpose we use **undistort** function from *OpenCV*, passing output parameters obteined from **calibrateCamera** function.

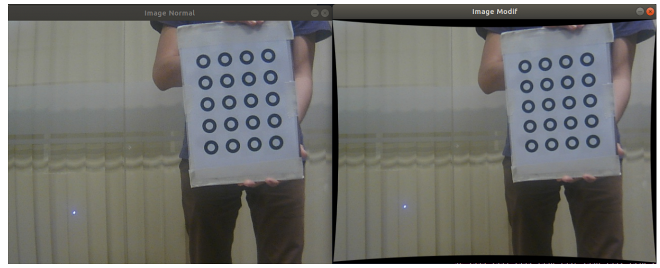

Figure 5. Image correction over concentric rings pattern.

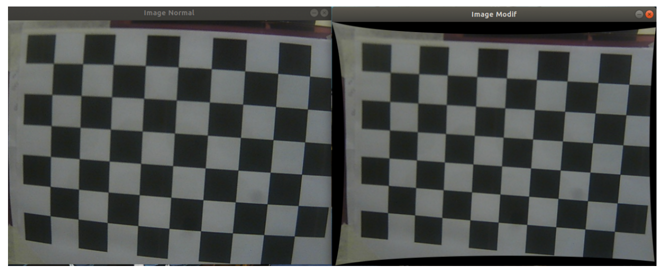

Figure 6. Image correction over chessboard pattern.

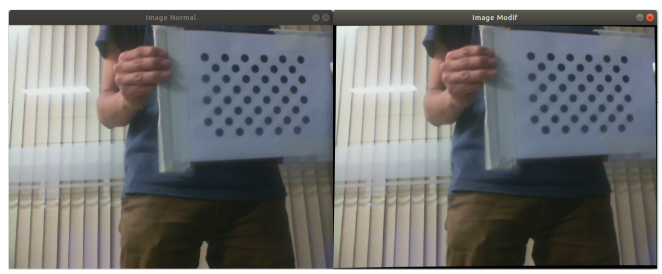

Figure 7. Image correction over assimetric disks pattern.

Another point to be considered is detected frames density (frames used for calibration), which allow us to see how is the distribution of detected frames over the scene, this is so important because an uniform distribution helps to get a good calibration, which implies that calibration parameters works well in whole scene. This distribution for assimetric disks is shown in figure 8.

3.4. Conclusions

Acknowledgments

This work was supported by grant 234-2015-FONDECYT (Master Program) from Cienciactiva of the National Council for Science,Technology and Technological Innovation (CONCYTEC-PERU).

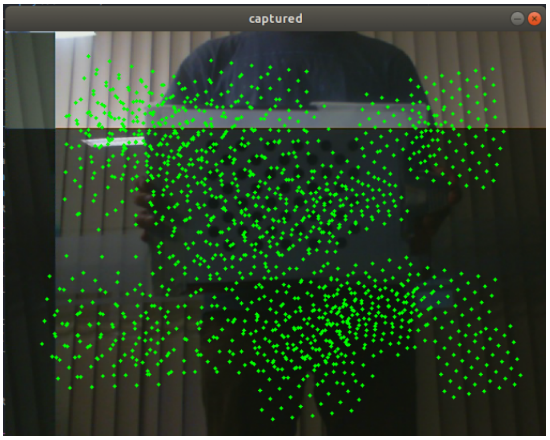

Figure 8. Detected pattern distribution on assimetric disks pattern.

References

[1] Zhang Z. (1998) A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research.

[2] Meier, Thomas, and King Ngi Ngan. "Automatic segmentation of moving objects for video object plane generation." IEEE Transactions on Circuits and Systems for Video Technology 8.5 (1998): 525-538.

[3] Cucchiara, Rita, et al. "Detecting moving objects, ghosts, and shadows in video streams." IEEE transactions on pattern analysis and machine intelligence 25.10 (2003): 1337-1342.

[4] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." Acm computing surveys (CSUR) 38.4 (2006): 13.

[5] Yang, Changjiang, Ramani Duraiswami, and Larry Davis. "Fast multiple object tracking via a hierarchical particle filter." Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 1. IEEE, 2005.

[6] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science and Business Media, 2010.