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.

Finally, after a good flat template tracking, some calculations, propper of calibration process are performed, this part gonna be explored in future presentations.

2. Method

Pipeline as is presented in 1, runs as follow:

1) Each frame of the image, it is first smoothed with median filter to obtain uniform black discs.
2) Smoothed frame then is turned in gray scale.
3) An elliptical kernel is applied for dilation to highlight the curves.
4) To highlight even more in some curves, we apply an adaptive threshold to counteract the light effects.
5) One more elliptical kernel for dilation is applied to get much more defined ellipses, also reduce discs impurities.
6) In binary image, we look for elliptic patters.
7) We obtain ellipses centers and connect them in order to stablish an orientation for tracking.

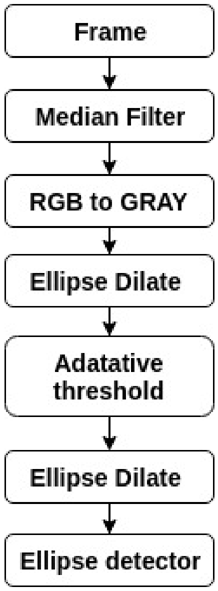

Figure 1. General pipeline

3. Results

In the following images, we present the resolts of our pipeline, getting the input image and final processed binary image near one to the other.

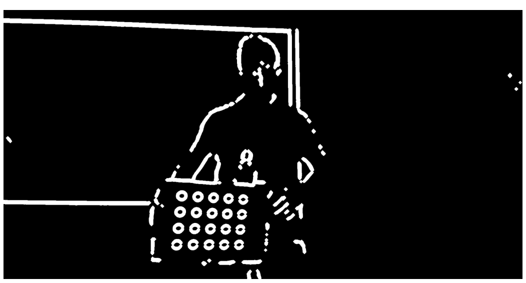

Figure 2. Final binary image in Kinect 2

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).

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.

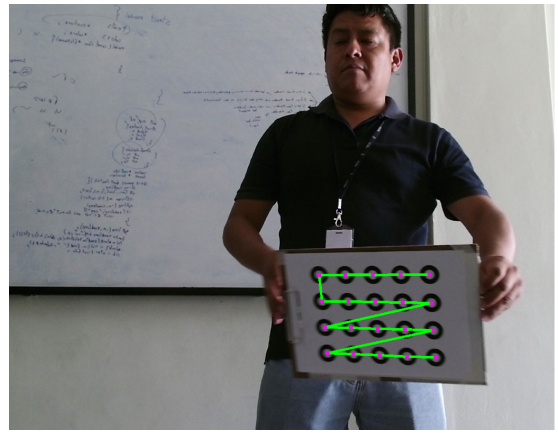

Figure 3. Input image of Kinect 2 and patter recognition

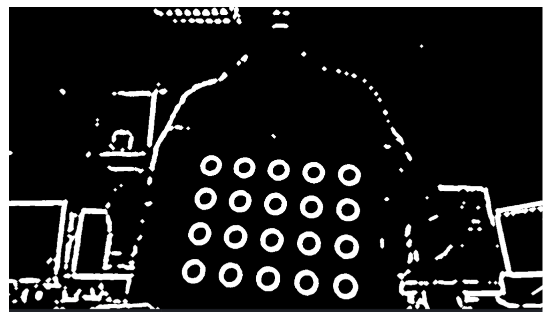

Figure 4. Final binary image in WebCam

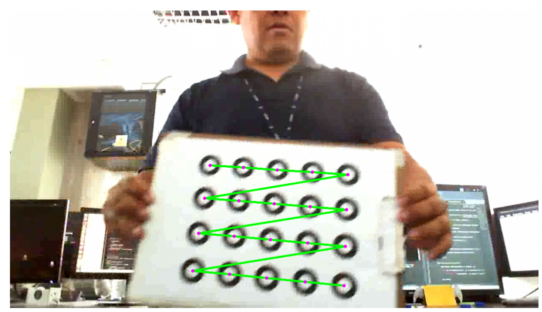

Figure 5. Input image of WebCam and patter recognition

[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.

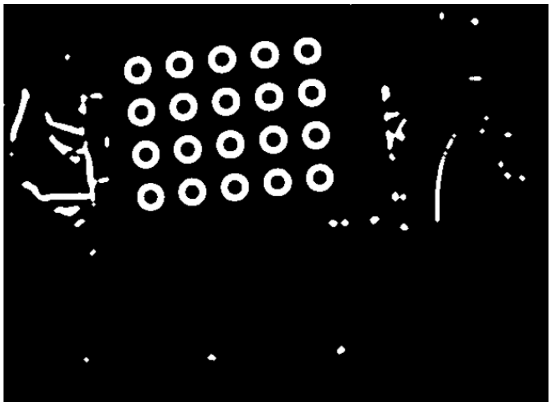

Figure 6. Final binary image in PS3

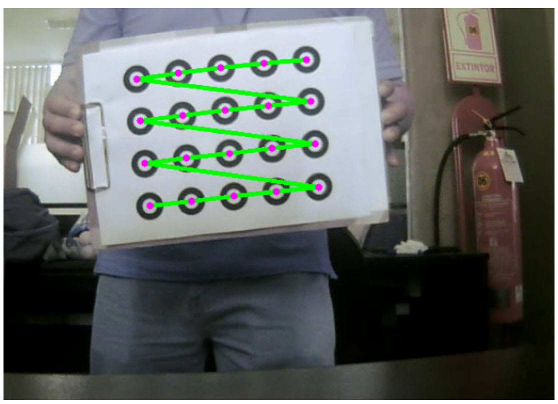

Figure 7. Input image of PS3 and patter recognition

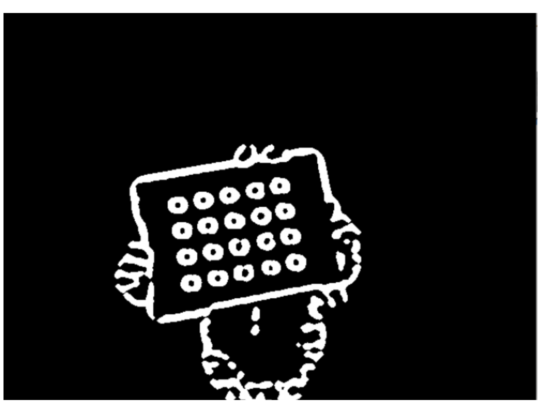

Figure 8. Final binary image in Intel deep map

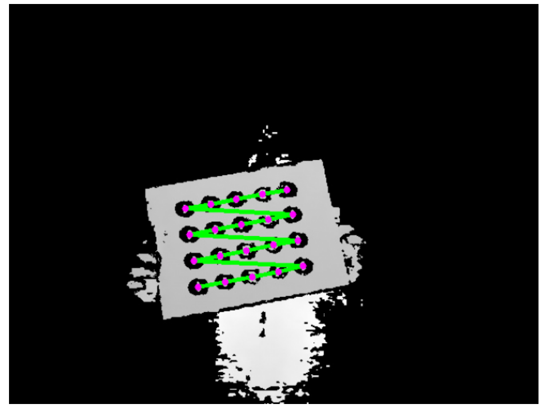

Figure 9. Input image of Intel deep map and patter recognition

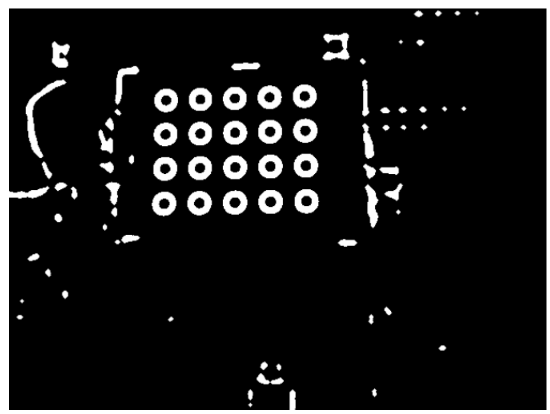

Figure 10. Final binary image in Intel RGB

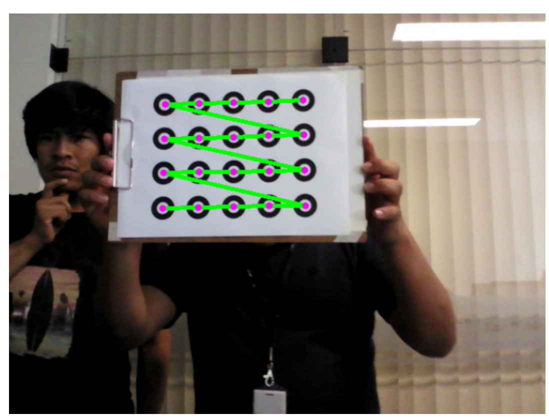

Figure 11. Input image of Intel RGB and patter recognition