Maestría en Ciencias de la Computación
Camera Calibration: Real-Time tracking of Control Points

Diego Alonso Javier Quispe , David Choqueluque Roman
Universidad Católica San Pablo
{ diego.javier, david.choqueluque}@ucsp.edu.pe
Arequipa,Perú

***Abstract*— The are many applications to machine vision, and is important to know the relation between the image of some object and its physical dimension in the space. The main idea to this work is obtain a good camera calibration procedure using a special grid determinate by circles. We developed the algorithm in c++ using OpenCV. We explain the Control Point Tracking algorithm for location of the control points in real time.**

Key words: Canny algorithm, Pattern tracking, Camera Calibration ,OpenCV

I. INTRODUCTION

An important step in the Camera Calibration pipeline is the detection and tracking of the control points of the pattern in real time. The objective of this step is to maintain a order of the control points detected through modifications in its position in space. In the present document we first explain the general pipeline that we follow, then we develop each stage of it.

II. GENERAL PIPELINE

The pipeline that we follow is composed of the following stages (Figure 1). First we read a video frame by frame, then we preprocess the frame, detect the control points and finally we follow up on them.

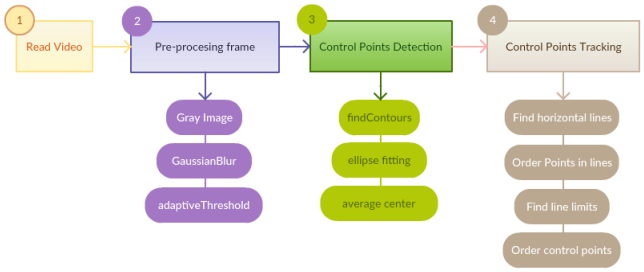

Fig. 1. General Pipeline.

III. STAGES

III-A. Read Video

In this stage we simply read frame-by-frame input video supported by Opencv reading functions.

III-B. Pre-processing Frame

In this stage we convert the input to gray scale, apply the Gaussian filter and finally perform the thresholding of the frame. The following images show the result of each operation.

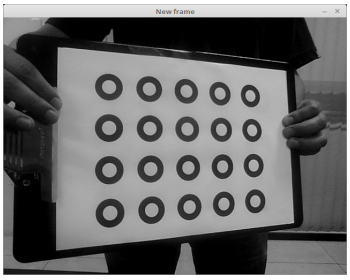

Fig. 2. Original Frame to GrayScale.

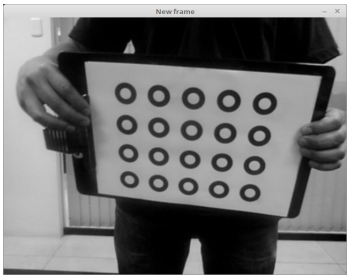

Fig. 3. Gaussian Filter.

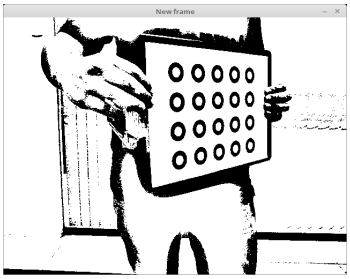

Fig. 4. Adaptative Threshold.

III-C. Control Point Detection

In this stage we takes as input the preprocesed image and look for contours, then we calculate the ellipse of these.

Whit the ellipses calculated above we select those that have a child and calculate the ellipse of this. To rule out false positive ellipses, we found that the centers of the parent-child ellipses do not exceed a maximum distance, this distance is calculated as half the radius of the ellipse son. The following image show the result of this stage. Finally for the ellipses that fulfill the condition we calculate the average of its two centers and keep it as a point of control of the pattern.

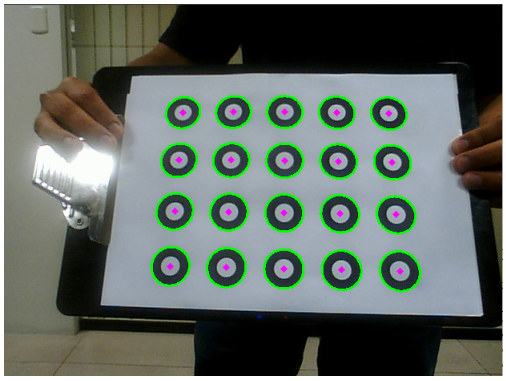

Fig. 5. Control Points Detection

III-D. Control Point Tracking

In this stage we take as input the control points calculated in the previous stage (Centers) and we proceed to find the horizontal lines (lines) that contain the centers of a row in the pattern. For that we take centers in pairs, calculate the equation of the line they define and for the remaining centers we calculate the distance from a point (center) to the previously calculated line, this is done until we obtain a line containing the number of ellipses in a row of the pattern (number of columns).

After calculating all the lines that the pattern contains, we proceed to enumerate each control point (center).

The following image shows the result of this stage.

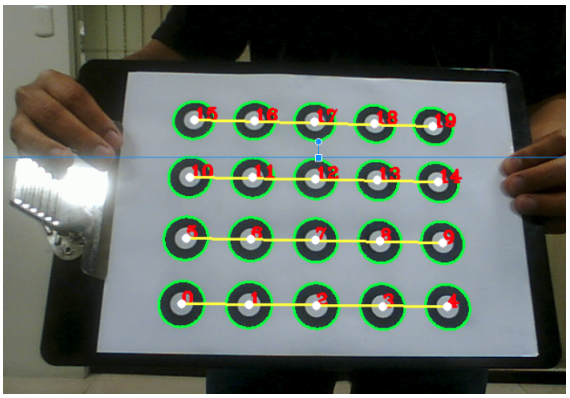

Fig. 6. Control Points Tracking.

IV. Experimental result

We have a general view of the four steps described in the pipeline, this window is shown below.

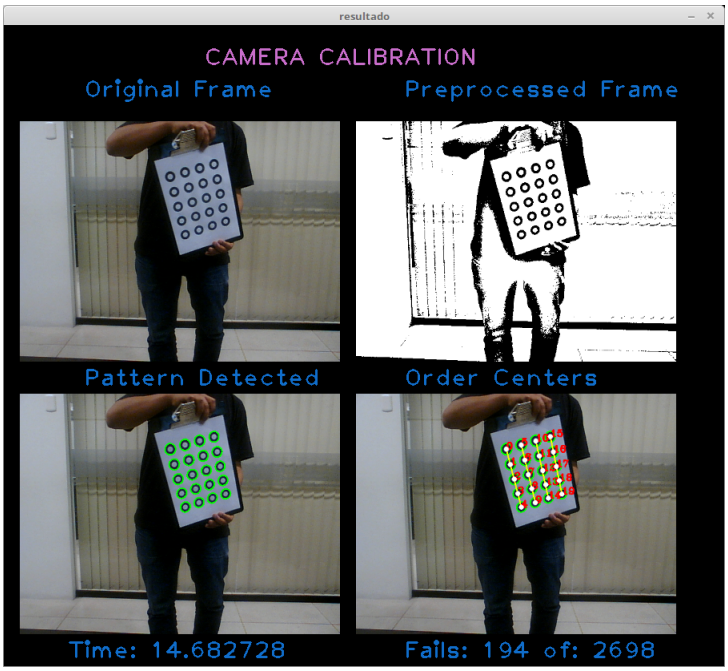

Fig. 7. General window to Camera Calibration.

We obtain the next result of out experiments:
Processing time to frame: 14.52 ms.
Accuracy: 86.59

References

[1] Zhengyou Zhang,"A Flexible New Technique for Camera Calibration", 2000.
[2] Ankur Datta, "Accurate Camera Calibration using Iterative Refinement of Control Points", 2009.
[3] Prakash, Çamera Calibration using Adaptive Segmentation and Ellipse Fitting for Localizing Control Points", 2012.