

# Camera Calibration using Levenberg-Marquardt Algorithm

Daniel Palomino  
dpalominop@gmail.com

**Abstract**—In several machine vision applications, a fundamental step is to precisely determine the relation between the image of the object and its physical dimension by performing a calibration process. The aim is to devise an enhanced mechanism for camera calibration in order to improve the already existing methods in OpenCV. A good calibration is important when we need to reconstruct a world model or interact with the world as in case of robot, hand-eye coordination. In the paper that i have based is proposed an enhanced camera calibration procedure using a special grid pattern of rings. The overall objective is to minimize the re-projection for good camera calibration.

## I. INTRODUCTION

A la hora de diseñar un sistema de Visión por Computador siempre se tienen en cuenta una serie de parámetros que serán decisivos para que la apariencia de los objetos en la imagen sea la mejor de cara a los posteriores algoritmos de análisis. Así se elegirá una óptica con una distancia focal,  $f$ , que permita que se observe con el suficiente tamaño en la imagen el elemento a buscar o que tenga una lente con la suficiente calidad como para que los objetos no se vean deformados. Respecto a la colocación de la cámara, ésta se realizará de forma que se perciba de la mejor manera posible el espacio por el que pueden aparecer los objetos. Para muchas aplicaciones estas medidas son suficientes.

Sin embargo, para aquellos algoritmos de Visión por Computador que necesiten extraer información 3D de una imagen o una secuencia de ellas o establecer la correspondencia entre dos o más cámaras, la calibración de los parámetros intrínsecos y extrínsecos del sistema de visión es una etapa fundamental.

## II. ALGORITMO ITERATIVO

El método desarrollado en el paper de Ankur Datta [1] se describe en los siguientes pasos:

### Objetivo

Dado  $N$  imágenes conteniendo una grilla plana de calibración, estimar los parámetros de la cámara.

### Algoritmo

1. **Detectar los puntos de Control:** Detectar los puntos de control en el patrón de calibración (esquinas, círculos o centros de anillos).
2. **Ajuste de Parámetros:** Usar los puntos de control detectados para estimar los parámetros de la cámara usando Levenberg-Marquardt.

Hacer hasta converger:

1. **Quitar la distorsión y la proyección:** Usar los parámetros de la cámara para quitar la distorsión y la proyección a las imágenes de entrada y obtener los padrones canónicos.
2. **Ubicar los puntos de control:** Ubicar los puntos de control en el padrón de calibración canónico.
3. **Reproyectar:** Proyectar los puntos de control usando los parámetros de la cámara estimados.
4. **Ajusta de parámetros:** Usar los puntos de control proyectados para refinar los parámetros de la cámara usando Levenberg-Marquardt.

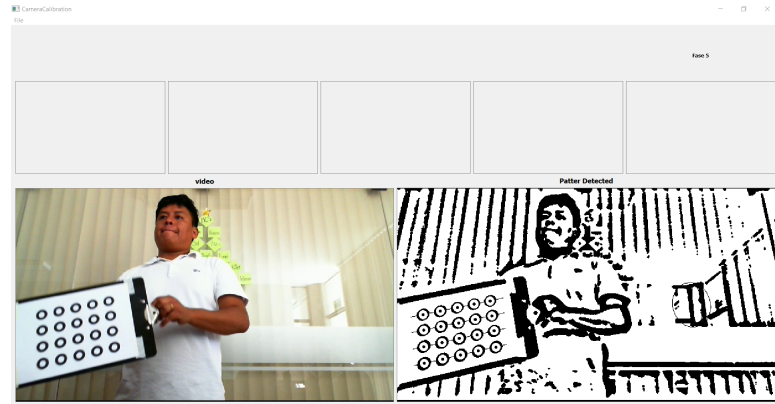


Fig. 1: Primera calibración de la cámara

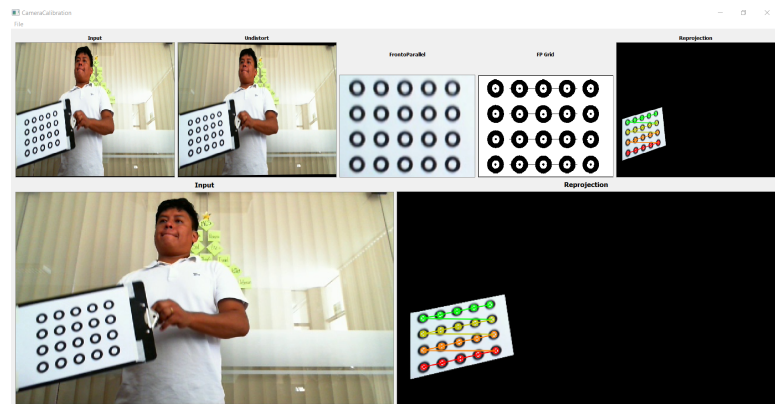


Fig. 2: Pipeline del algoritmo iterativo

### III. RESULTADOS

El algoritmo final ha sido probado usando las siguientes cámaras:

1. LiveCam Microsoft.
2. PlayStation2 Cam.

#### III-A. LiveCam Microsoft

Resultados usando un algoritmo iterativo simple con 30 frames y 30 iteraciones:

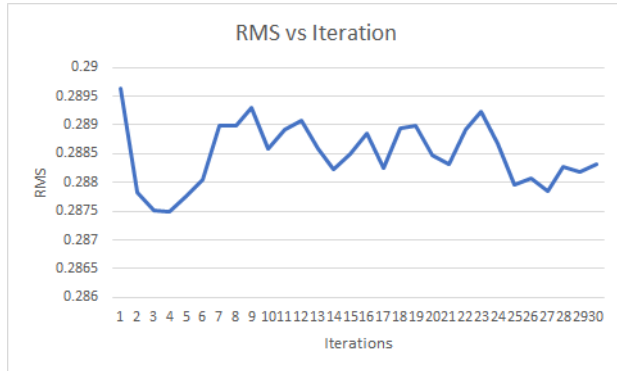


Fig. 3: RMS obtenido

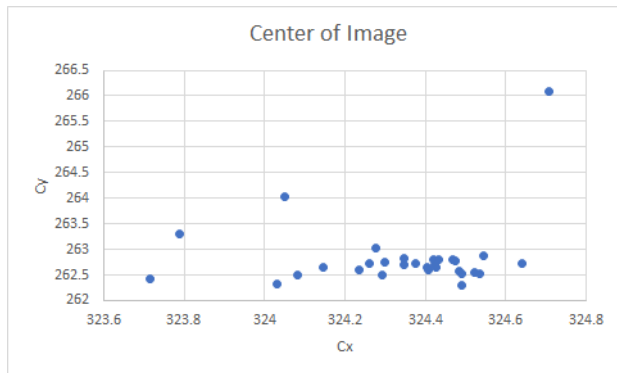


Fig. 4: Distribución de centros obtenido

Resultados usando el algoritmo de Levenberg-Marquardt con 30 frames y 14 iteraciones:

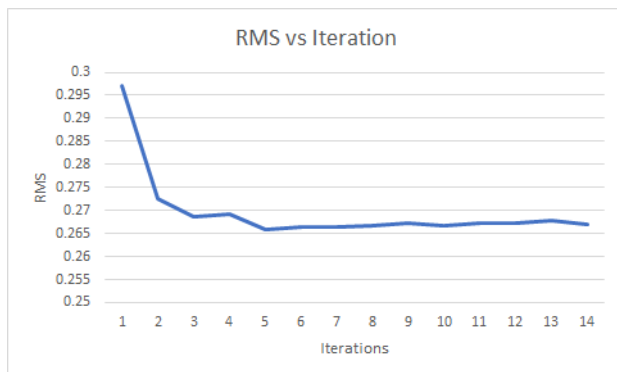


Fig. 5: RMS obtenido

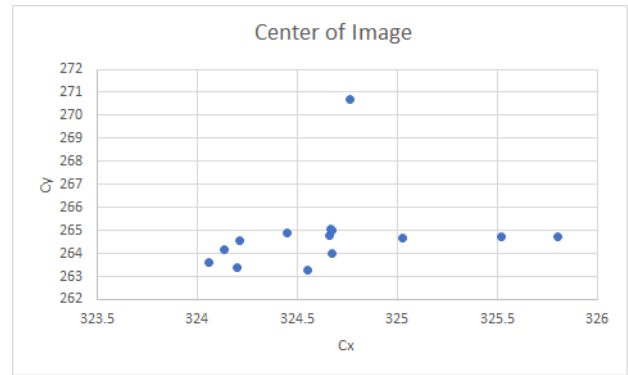


Fig. 6: Distribución de centros obtenido

De los gráficos anteriores se puede apreciar que con el algoritmo de Levenberg-Marquardt se llega a una convergencia en un número menor de iteraciones además de un RMS muchos más bajo.

#### III-B. PlayStation2 Cam

Resultados usando un algoritmo iterativo simple con 70 frames y 30 iteraciones:

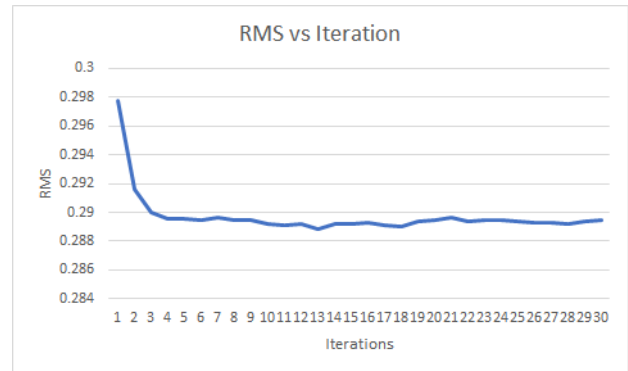


Fig. 7: RMS obtenido

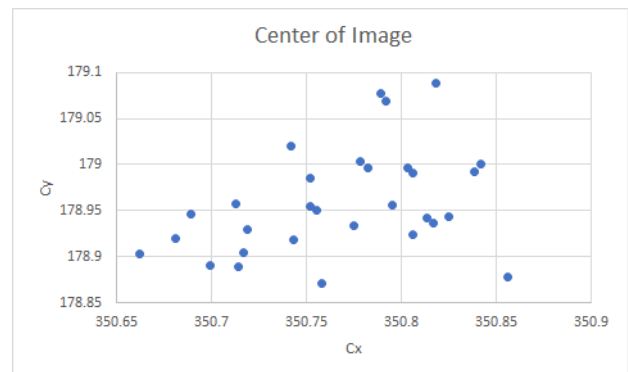


Fig. 8: Distribución de centros obtenido

Resultados usando el algoritmo de Levenberg-Marquardt con 70 frames y 10 iteraciones:

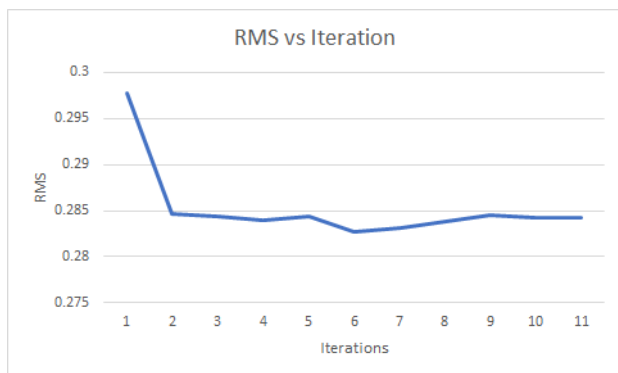


Fig. 9: RMS obtenido

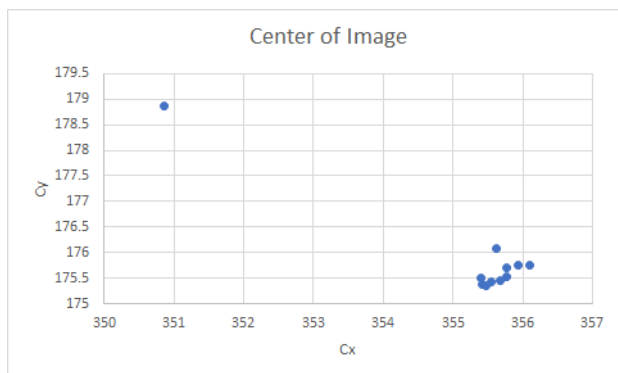


Fig. 10: Distribución de centros obtenido

Al igual que en el caso anterior, de los gráficos anteriores se puede apreciar que con el algoritmo de Levenberg-Marquardt se llega a una convergencia en un número menor de iteraciones además de un RMS muchos más bajo.

#### IV. REDUCCIÓN DE LA DISTORSIÓN EN LAS IMÁGENES

Usando la cámara **LiveCam Microsoft**:



Fig. 11: Imagen inicial



Fig. 12: Imagen Final

Usando la cámara **PlayStation2 Cam**:



Fig. 13: Imagen inicial



Fig. 14: Imagen Final

#### REFERENCES

- [1] A. Datta, J. Kim and T. Kanade, "Accurate camera calibration using iterative refinement of control points," 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops (ICCV WORKSHOPS), Kyoto, 2010, pp. 1201-1208. doi:10.1109/ICCVW.2009.5457474
- [2] Asthana, Shubham. (2014). Enhanced Camera Calibration for Machine Vision using OpenCV. International Journal of Artificial Intelligence. Volume 3.
- [3] <https://docs.opencv.org/>