

Prototipo de motor de Realidad Aumentada tridimensional para dispositivos móviles

Xoan Iago Suárez Canosa

Máster Universitario en Inteligencia Artificial
Julio 2016

PCR

2. Metodología

Índice

1. **Introducción**

2. Metodología

1. Calibración

1. El problema
2. Estado del Arte
3. Solución propuesta: Calibración automática
4. Conversión a OpenGL
5. Experimentos realizados

2. Localización de letreros

1. El problema
2. Estado del Arte
3. Solución propuesta
4. Experimentos realizados

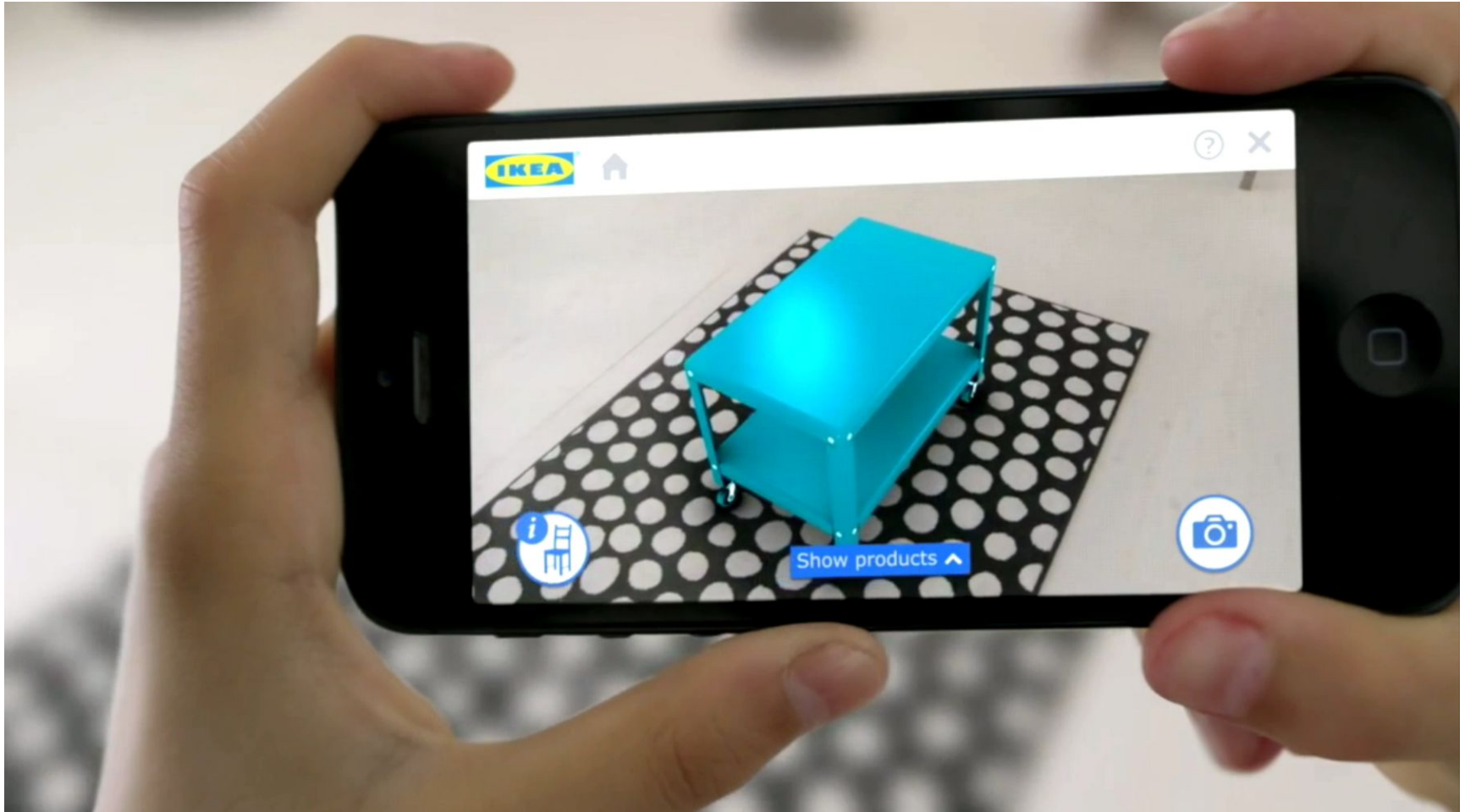
3. Consideraciones Ingenieriles

4. Demostración

3. Conclusiones y trabajos futuros

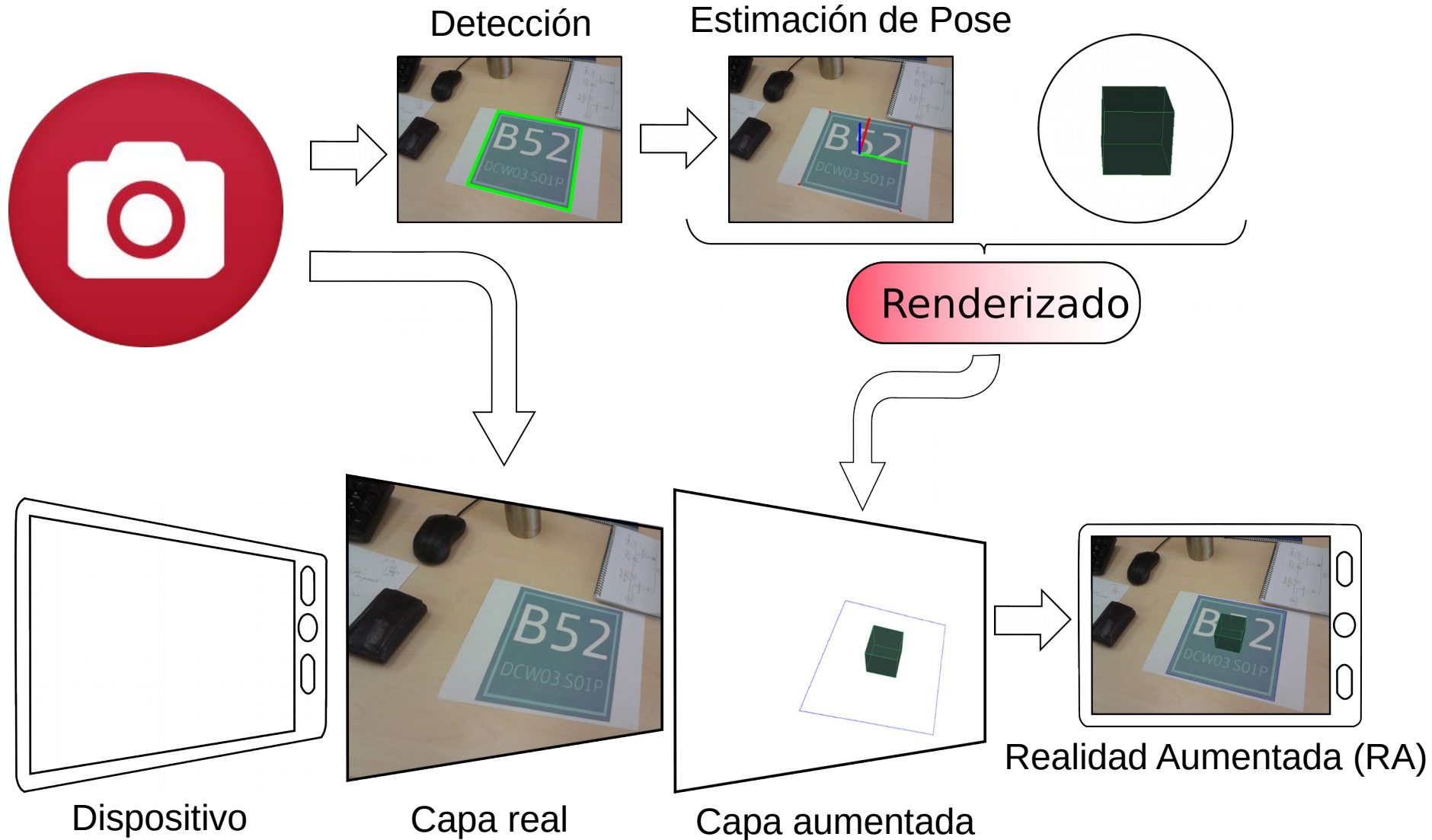
4. Referencias

1. Introducción



Introducción

1.2 ¿Cómo funciona una aplicación de RA?



“Desarrollar una aplicación para realizar Realidad Aumentada en dispositivos móviles sobre letreros”

¿Qué es un letrero?

- Aplicación de Gestión de salas
- Identificador de salas



Problemas:

- Procedimiento de calibración tedioso para usuarios móviles
- Detección de objetos sin textura

- Desarrollar aplicación para realizar Realidad Aumentada en dispositivos móviles sobre letreros.
- Localizar los letreros en las imágenes de la cámara y calcular la posición en base a él.
- Mostrar objetos gráficos virtuales sobre el letrero de forma precisa.



2. Metodología

2. Metodología

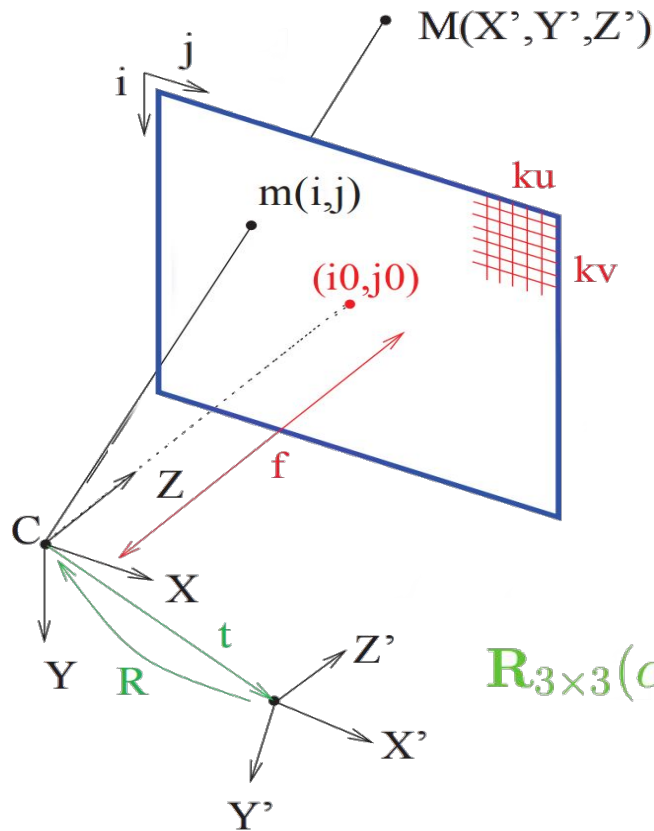
Índice

1. Introducción
2. Metodología
 - 1. Calibración**
 1. El problema
 2. Estado del Arte
 3. Solución propuesta: Calibración automática
 4. Conversión a OpenGL
 5. Experimentos realizados
 2. Localización de letreros
 1. El problema
 2. Estado del Arte
 3. Solución propuesta
 4. Experimentos realizados
 3. Consideraciones Ingenieriles
 4. Demostración
3. Conclusiones y trabajos futuros
4. Referencias

Metodología

2.1 Calibración

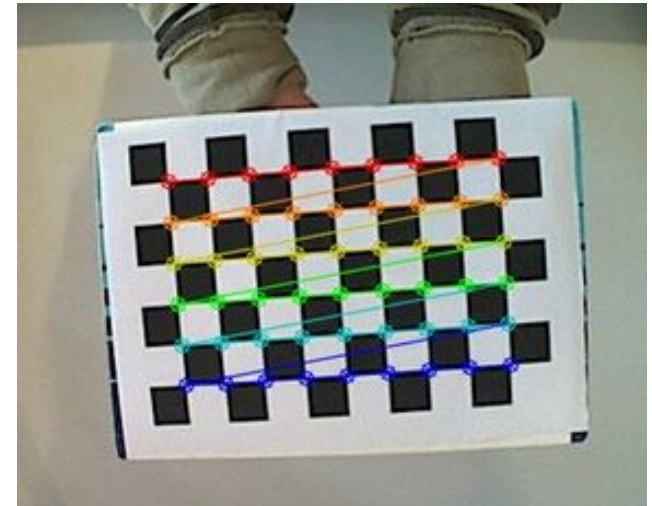
- Parámetros Intrínsecos (**K**)
- Parámetros Extrínsecos (**R|t**)



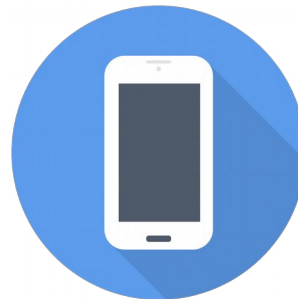
$$\begin{pmatrix} \lambda j \\ \lambda i \\ \lambda \end{pmatrix} = \underbrace{\mathbf{K} (\mathbf{R}_{3 \times 3} \mid \bar{t}_{3 \times 1})}_{\mathbf{P}_{3 \times 4}} \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix}$$

$$\mathbf{R}_{3 \times 3}(\alpha, \beta, \gamma), \quad \bar{t}_{3 \times 1} = \begin{pmatrix} t_X \\ t_Y \\ t_Z \end{pmatrix} \quad \mathbf{K} = \begin{pmatrix} f k_u & s & j_0 \\ 0 & f k_v & i_0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Los intrínsecos de la cámara(**K**) son imprescindibles para hacer RA ya que se utilizan para **estimar la pose** y **proyectar los puntos** del modelo virtual 3D en la imagen.
- El proceso de calibración resulta tedioso para usuarios de dispositivos móviles.
- La calibración puede no realizarse adecuadamente.

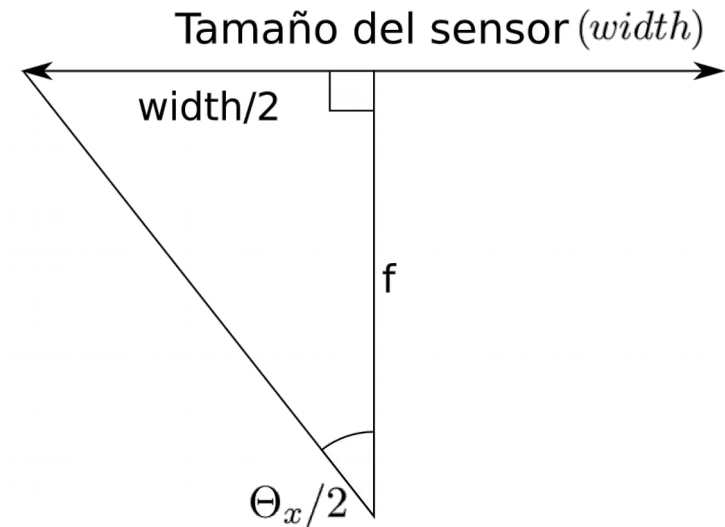
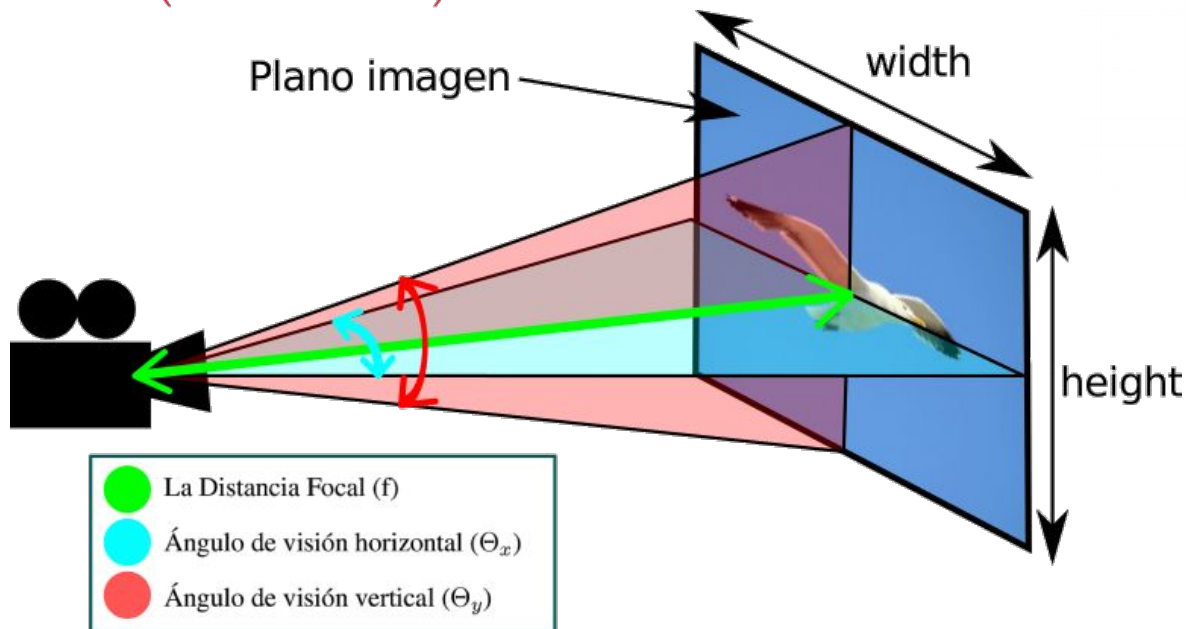


- Suposición de Intrínsecos fijos
- BD de Intrínsecos
- Auto-calibración



- Calibración automática en base a los parámetros físicos de la cámara:

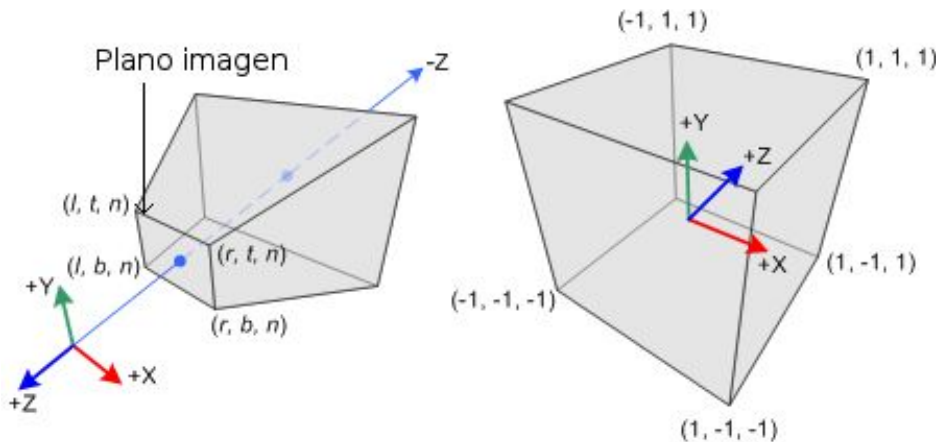
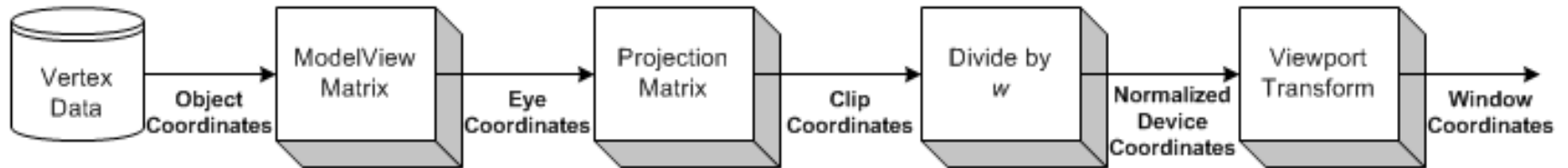
$$\mathbf{K} = \begin{pmatrix} f k_u & s & j_0 \\ 0 & f k_v & i_0 \\ 0 & 0 & 1 \end{pmatrix}$$



$$\tan(\Theta_x/2) = \frac{width/2}{f}$$

$$width/2 = f \cdot \tan(\Theta_x/2)$$

$$width = 2f \cdot \tan(\Theta_x/2)$$

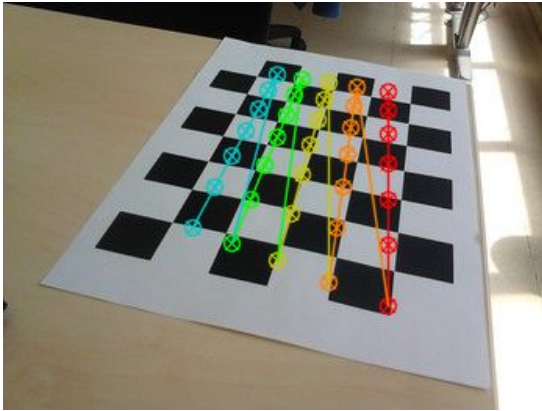


- $r - l \Leftrightarrow width$
- $r + l \Leftrightarrow i_0 - (width/2)$
- $t - b \Leftrightarrow height$
- $t + b \Leftrightarrow -(j_0 - (height/2))$

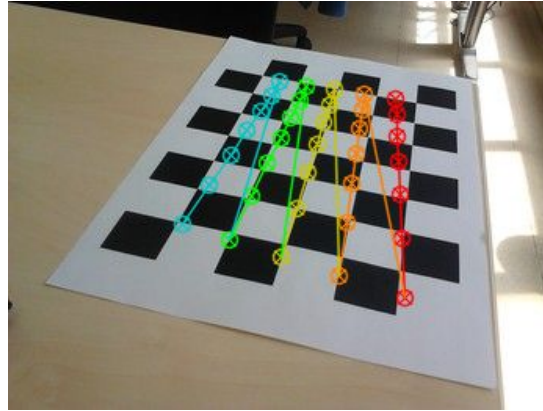
- $P_{11} = \frac{2n}{r-l} \Leftrightarrow \frac{2fk_u}{width}$

- $P_{22} = \frac{2n}{t-b} \Leftrightarrow \frac{2fk_v}{height}$

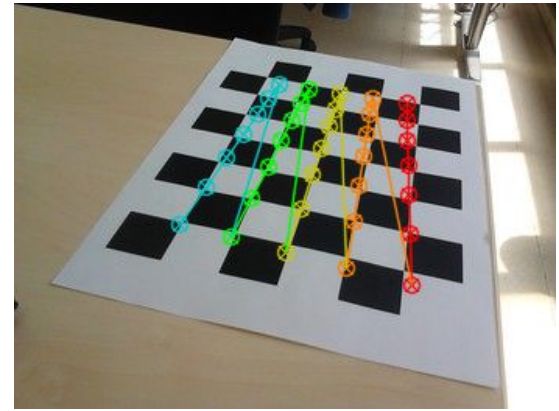
- Reproyección de las esquinas con distintas f :



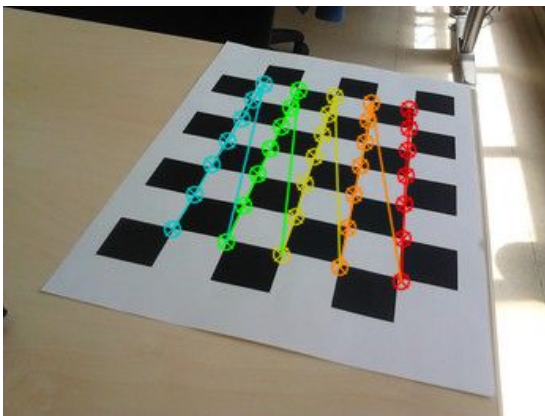
$f = 10\%$



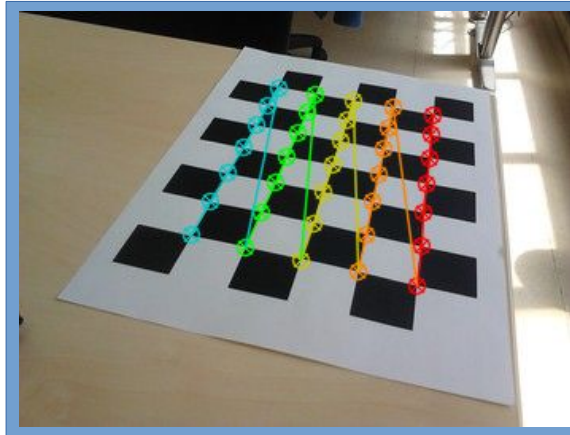
$f = 30\%$



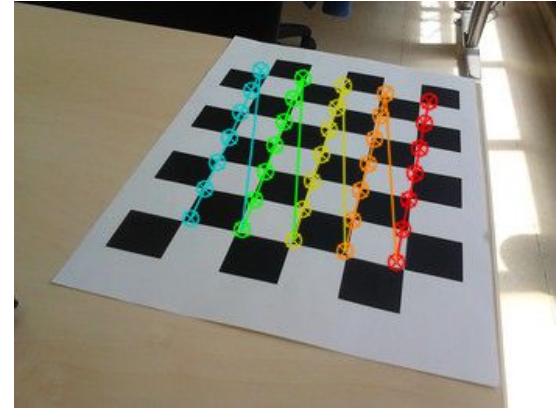
$f = 50\%$



$f = 80\%$



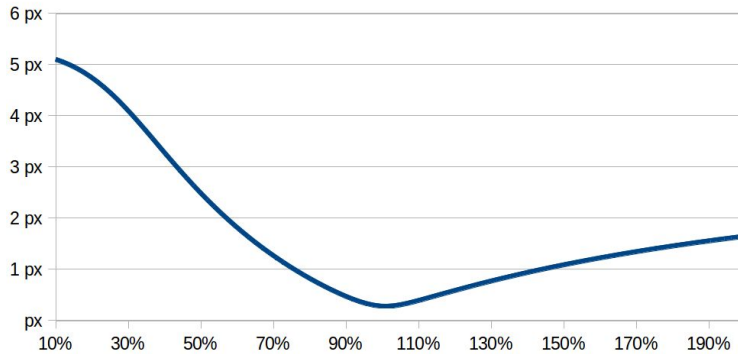
$f = 100\%$



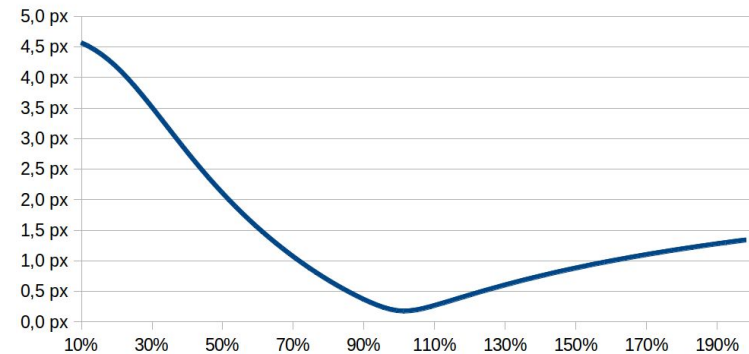
$f = 200\%$

- Reproyección de las esquinas con distintas f :

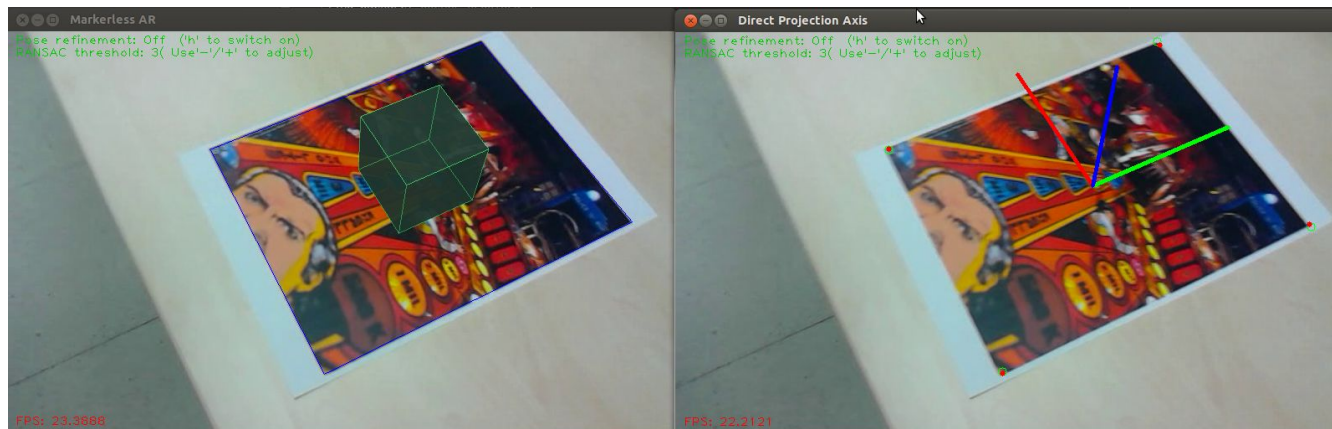
Huawei U8650 (2048x1536px)



Motorola Moto X (1920x1080px)



- Procesamiento de secuencias y Realidad Aumentada con la proyección de OpenGL:



2. Metodología

Índice

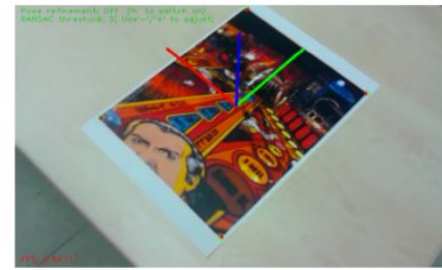
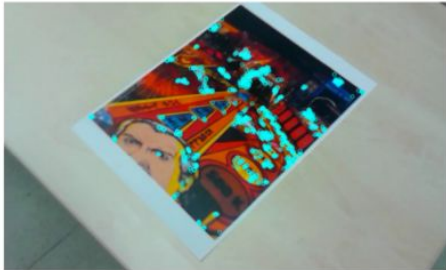
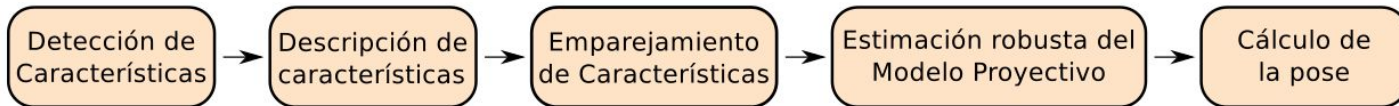
1. Introducción
2. Metodología
 1. Calibración
 1. El problema
 2. Estado del Arte
 3. Solución propuesta: Calibración automática
 4. Conversión a OpenGL
 5. Experimentos realizados
 - 2. Localización de letreros**
 1. El problema
 2. Estado del Arte
 3. Solución propuesta
 4. Experimentos realizados
 3. Consideraciones Ingenieriles
 4. Demostración
3. Conclusiones y trabajos futuros
4. Referencias

- Queremos usar letreros como patrón sobre el que mostrar imágenes en Realidad Aumentada.
- Estos letreros letreros son visualmente estéticos e informativos.
- Esta aplicación formará parte de un sistema de gestión de salas.



- Hemos de localizar el letrero para poder estimar la posición de la cámara.
- Los letreros no contiene textura, por lo que no es posible emplear las técnicas clásicas para localizarlos en un fotograma.
- Dado que se conocen de antemano los colores y la forma del letrero puede emplearse esta información para encontrarlo.

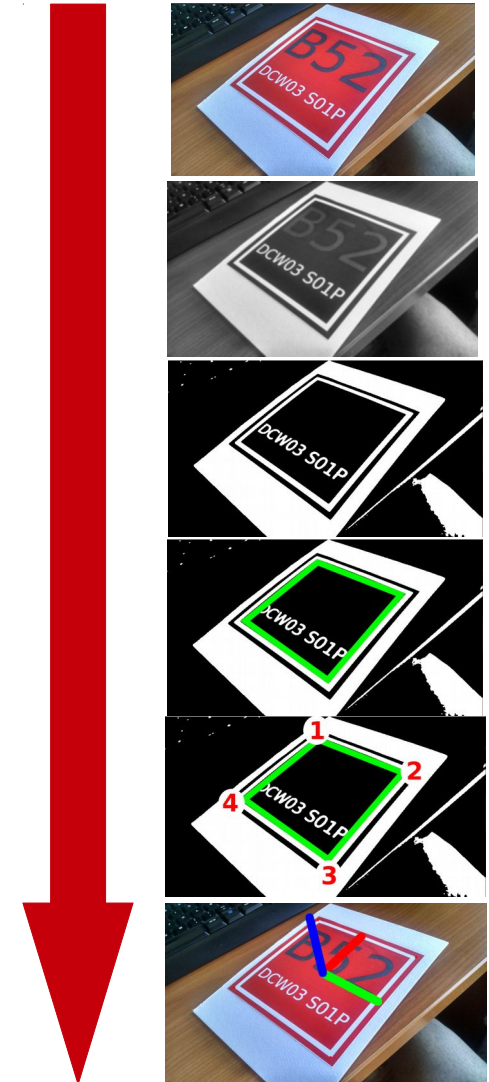
- Métodos basados en características



- Métodos Directos
- Métodos basados en Marcadores



- Flujo de trabajo
 - Obtención de la imagen
 - Paso a escala de grises
 - Binarización
 - Selección de contornos
 - Ordenación de esquinas
 - Estimación de pose



- Búsqueda en el espacio de color

Se emplea la conversión a escala de grises que maximiza el contraste entre los dos colores del letrero.



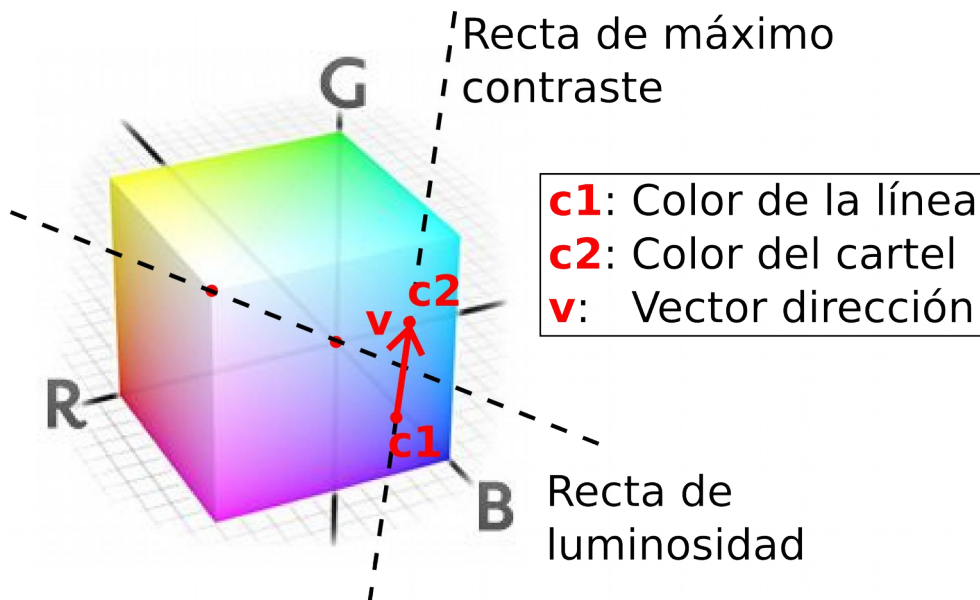
Original



Escala de Grises



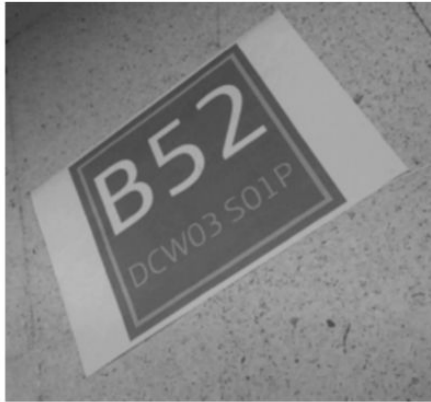
Salida del Algoritmo



Localización de letreros

2.2.3 Solución propuesta

- Binarización



(a) Escala de grises



(b) Contornos Canny



(c) Umbral = 100 (Nivel 0)



(d) Umbral = 101 (N. 1 hacia arriba)



(e) Umbral = 99 (N. 1 hacia abajo)



(f) Umbral = 103 (N. 2 hacia arriba)

- Selección de contornos

- 1) Simplificación de contornos

Algoritmo Ramer-Douglas-Peucker

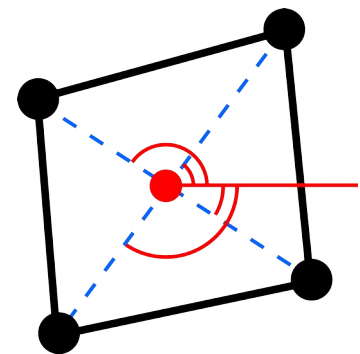
- 2) Selección de contornos cuadrados

- 3) Selección de contornos con un área mayor que *minArea*.

- 4) Selección de contornos convexos

- 5) Seleccionar pares de candidatos que formen una línea, para ello comparamos que sus centros de gravedad sean cercanos.

- 6) Ordenación de los puntos



Localización de letreros

2.2.4 Experimentos realizados

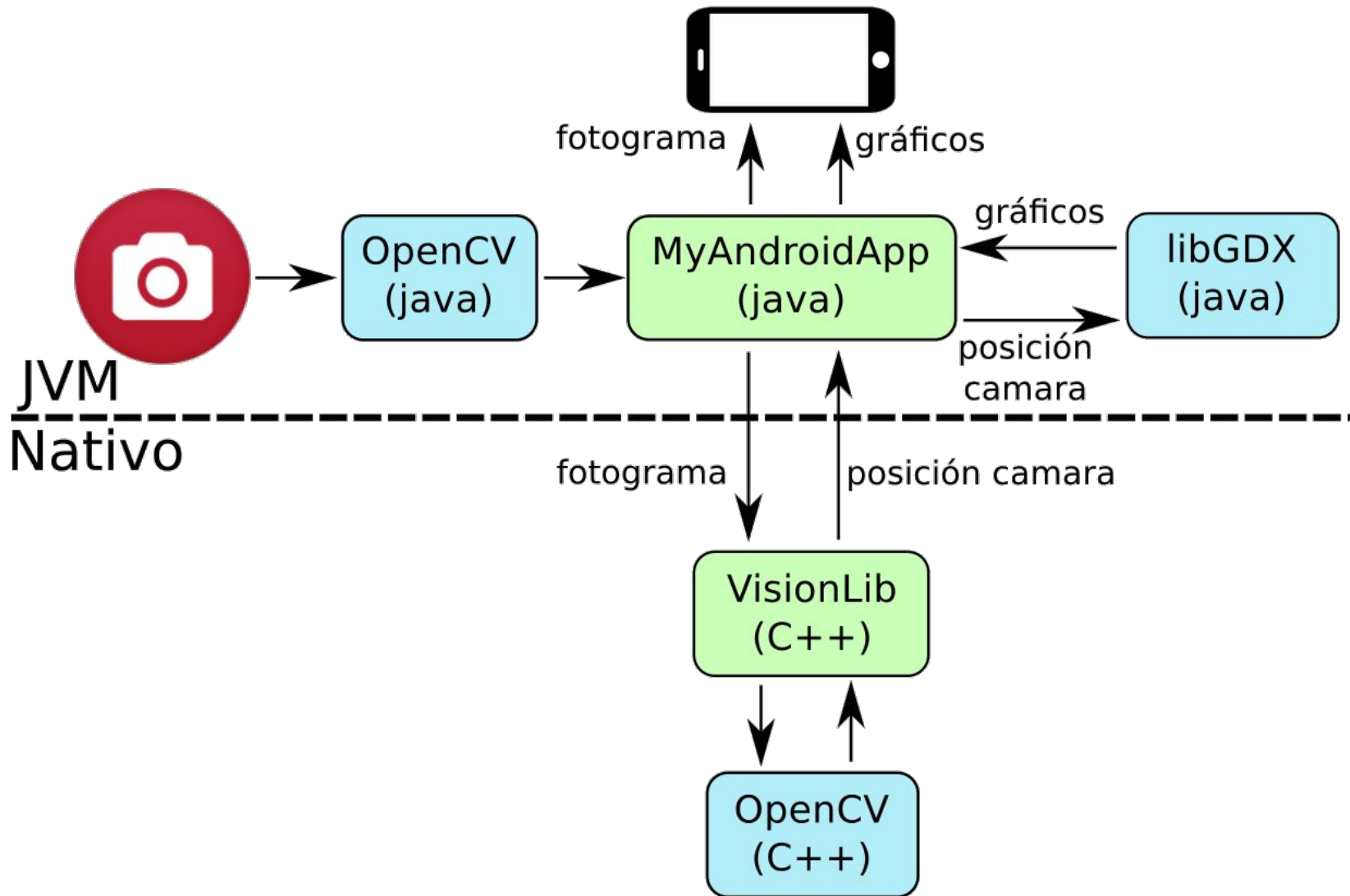
- Se han procesado secuencias de imágenes sometidas a:
 - Variaciones perspectivas
 - Distintas condiciones de iluminación
 - Ruido
 - Distintos colores de cartel



2. Metodología

Índice

1. Introducción
2. Metodología
 1. Calibración
 1. El problema
 2. Estado del Arte
 3. Solución propuesta: Calibración automática
 4. Conversión a OpenGL
 5. Experimentos realizados
 2. Localización de letreros
 1. El problema
 2. Estado del Arte
 3. Solución propuesta
 4. Experimentos realizados
 - 3. Consideraciones Ingenieriles**
 4. Demostración
3. Conclusiones y trabajos futuros
4. Referencias



- Aplicación Android (Java + C++)
- Librería de visión: OpenCV
- Motor gráfico:
 - LibGDX (Demo Android)
 - OpenGL (Experimentos PC)
- Metodología Ágil SCRUM adaptada a 1 desarrollador. Herramientas (A.S., GitHub, RedMine)
- Tiempo de desarrollo: 5 meses (600h)
- Coste estimado: 7800€

2. Metodología

Índice

1. Introducción
2. Metodología
 1. Calibración
 1. El problema
 2. Estado del Arte
 3. Solución propuesta: Calibración automática
 4. Conversión a OpenGL
 5. Experimentos realizados
 2. Localización de letreros
 1. El problema
 2. Estado del Arte
 3. Solución propuesta
 4. Experimentos realizados
 3. Consideraciones Ingenieriles
 - 4. Demostración**
3. Conclusiones y trabajos futuros
4. Referencias

Metodología

2.4 Demostración



3. Conclusiones y trabajo futuro

2. Metodología

Índice

1. Introducción
2. Metodología
 1. Calibración
 1. El problema
 2. Estado del Arte
 3. Solución propuesta: Calibración automática
 4. Conversión a OpenGL
 5. Experimentos realizados
 2. Localización de letreros
 1. El problema
 2. Estado del Arte
 3. Solución propuesta
 4. Experimentos realizados
 3. Consideraciones Ingenieriles
 4. Demostración
- 3. Conclusiones y trabajos futuros**
4. Referencias

3. Conclusiones y trabajos futuros

- Se ha alcanzado el objetivo planteado mediante la construcción de un sistema de RA tridimensional para dispositivos móviles.
- Se han hecho además 2 contribuciones teóricas:
 - Método para realizar la calibración automática
 - Algoritmo para detección de letreros
- Trabajos futuros:
 - Tracking
 - Distorsiones ópticas
 - Sensores
 - Oclusiones
 - Patrones tridimensionales
 - Múltiples patrones
 - Otros SO's
 - ...

4. Referencias

2. Metodología

Índice

1. Introducción
2. Metodología
 1. Calibración
 1. El problema
 2. Estado del Arte
 3. Solución propuesta: Calibración automática
 4. Conversión a OpenGL
 5. Experimentos realizados
 2. Localización de letreros
 1. El problema
 2. Estado del Arte
 3. Solución propuesta
 4. Experimentos realizados
 3. Consideraciones Ingenieriles
 4. Demostración
3. Conclusiones y trabajos futuros
- 4. Referencias**

4. Referencias

- Engel, J., Schöps, T., & Cremers, D. (2014, September). LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision* (pp. 834-849). Springer International Publishing.
- Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge university press.
- Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3), 244-256.
- Baumela, L. (2015). Tema 4. Modelado de cámara y calibración. *Visión por computador*.



¡Gracias!