

PROYECTO DE FIN DE ESTUDIOS
EN LA CARRERA INGENIERÍA ELÉCTRICA

encuadro

Aplicación de realidad aumentada y navegación para museos sobre
dispositivos móviles

DOCUMENTACIÓN SEGUNDO HITO
AÑO 2012

Tutor:

Juan Cardelino

Integrantes:

Juan Braun

juanibraun@gmail.com

Martín Etchar

mrtn.etchart@gmail.com

Pablo Flores

pablofloresguridi@gmail.com

Mauricio González

mgonzaleznapa@gmail.com

Índice general

Índice general	1
1. Introducción	2
1.1. Sobre el documento	2
1.2. Entregables planificados para la fecha del segundo hito	2
2. Navegación	4
2.1. Navegación con <i>access points</i>	4
2.2. Navegación con GPS	4
2.3. Navegación con códigos QR	4
3. Realidad aumentada	6
3.1. Definición	6
3.2. Fundamento	6
3.3. Modelado 3D	7
3.4. Rendering	7
4. Estimación de pose monocular	8
4.1. Calibración de cámara: modelo pin-hole [1]	8
4.1.1. Fundamentos y definiciones	8
4.1.2. Matriz de proyección	10
4.2. Algoritmos de extracción de características	12
4.2.1. Segmentos	12
4.3. Marcadores	12
4.3.1. Marcador QR	12
4.3.1.1. Elementos estructura del marcador	13
4.3.1.2. Diseño	14
4.3.1.3. Filtrado de segmentos	15
4.3.1.4. Determinación de correspondencias	15
4.4. Algoritmos de estimación de pose monocular	15
4.5. POSIT	15
4.5.1. Hipotesis de trabajo	15
4.5.2. Problema a resolver	16
4.5.3. Proyeccion ortografica escalada (SOP)	16
4.5.4. Ecuaciones fundamentales	17
4.6. validación de algoritmos	17
Bibliografía	19

CAPÍTULO 1

Introducción

1.1. Sobre el documento

Este trabajo cuenta con la documentación del Proyecto *enCuadro*, en el marco del proyecto de fin de carrera, para la fecha del segundo hito. En el mismo, se encuentra una comparación entre lo que se planteaba tener para la fecha durante la planificación y lo que se tiene realmente ahora; además de un cronograma con las actividades a desempeñar a lo largo de los meses que quedan para poder cumplir con los objetivos.

Es conveniente recordar al lector que este proyecto cuenta con tres partes fundamentales:

- Navegación
- Identificación de obras
- Realidad aumentada

Si bien la mayor parte de las energías se han enfocado en la realidad aumentada, las dos primeras partes son importantes en lo que refiere a la completitud del desarrollo de la aplicación.

En este documento se hace también una descripción detallada de las conclusiones obtenidas hasta el momento en cuanto al sistema de navegación que se eligió, los algoritmos para identificar las obras que se han estudiado, las herramientas de *rendering* utilizadas para la realidad aumentada y los algoritmos de estimación de pose que se han implementado y comparado. Se presentan además los casos de uso que aspiramos llegar a implementar.

1.2. Entregables planificados para la fecha del segundo hito

A continuación se listan los trabajos que se esperaba tener prontos cuando se realizó la planificación, en octubre de 2011.

- Implementación del algoritmo de navegación
- Implementación del algoritmo de realidad aumentada
- Implementación del algoritmo de identificación
- Concatenación de los tres bloques anteriores

- Prueba del programa

A lo largo de este documento se detallará el estado de cada uno de ellos.

CAPÍTULO 2

Navegación

Se estudiaron distintas alternativas para la navegación o localización en el interior del museo. La primera posibilidad analizada fue la utilización de tres o más *access points*, mediante los cuales, una vez mapeadas las características de las señales en cada uno de los puntos de las salas, se podría ubicar al usuario dentro de las mismas. Otra forma de navegación que se tuvo en cuenta fue la localización a través de la tecnología GPS. Sin embargo, se optó por utilizar códigos QR dada su amplia difusión, practicidad y facilidad de implementación.

2.1. Navegación con *access points*

Al igual que la localización mediante la tecnología GPS, la localización interior con estándares IEEE 802.11 (WiFi), se obtiene mediante el procesamiento de señales RF. La idea es, una vez conocida la posición de los transmisores WiFi (*Access Points*) presentes en una sala, se miden las distintas señales recibidas por un dispositivo móvil (provinientes de cada uno de los APs). De esta manera, se logra ubicar al móvil dentro de la sala. Es posible utilizar diferentes medidas, como por ejemplo, la potencia de las señales recibidas, el tiempo de propagación de cada una de ellas o incluso sus ángulos de incidencia.

Debido de su gran complejidad, esta alternativa se descartó casi inmediatamente, ya que al idea no es centrar el proyecto en la navegación de interiores, sino usarla simplemente como una herramienta para la completitud de la aplicación final.

2.2. Navegación con GPS

La tecnología GPS (*Global Positioning System*), es altamente usada como sistema asesor para vehículos y sistemas de información geográfica. Algunas bibliografías afirman que su precisión supera los 3m en el 95 % de los casos [3]. Sin embargo, resulta imposible utilizar esta tecnología en lugares cerrados o muy edificadas. Se decidió entonces que esta no era la herramienta indicada para el problema en cuestión.

2.3. Navegación con códigos QR

Los códigos QR (*Quick Response*) son algo así como códigos de barras en dos dimensiones creados por la compañía japonesa *Denso Wave* en 1994 [2]. En ellos, se puede almacenar una cantidad relativamente alta de información, la cual puede ser leída a gran velocidad. Resulta muy

común hoy día que los dispositivos móviles cuenten con lectores de códigos QR, los que por lo general derivan al usuario a una dirección web determinada. Para el caso de nuestra aplicación en particular, estos códigos podrían ubicarse en las distintas salas del museo y mediante la lectura de los mismos (habría que incorporar esta funcionalidad a la aplicación), se podría saber en qué sala se encuentra el usuario e incluso qué cuadros de interés se encuentran en la misma.

Para incorporar navegación con códigos QR en la aplicación se requiere entonces solucionar los siguientes problemas:

- Codificar y decodificar los códigos QR.
- Generar una base de datos (por red o local) con información de la sala.
- Acceder a la base de datos desde el dispositivos de manera sencilla y ágil.

Para generar éstos códigos, existen formas ya implementadas en las que en forma muy sencilla se ingresa un texto, un url o una localización geográfica relativa a googlemaps y se obtiene el QR correspondiente. Tal es el caso del sitio <http://www.qrcodegener.com/>, que fue utilizado para generación de la base de datos de códigos QR implementada hasta el momento.

En cuanto a la decodificación, existen distintas alternativas que pueden ser programadas en C/C++ o Java. En C/C++ se estudió el método de Fackuchi que permite decodificar distintos tipos de QR con distintos tamaños y cantidad de información. Fue desarrollado utilizando la librería OpenCV. En Java se estudió el contenido de la librería Zxing, de código abierto, que permite decodificar muy bien. Sin embargo, debido a que ya existen librerías que trabajan sobre Xcode (el kit de desarrollo para sistemas IOS) que implementan funcionalidades para decodificar este tipo de códigos, se optó por estudiar más a fondo estas. Las mismas son ZBar y Zxing-iPhone.

Se implementó una aplicación utilizando el Framework Zbar, que al presionar un botón lee el QR, lo decodifica y luego despliega su información. Para probar la aplicación, se codificó un texto estándar con el mensaje: “Usted se encuentra en la zona de cuadros de *NombreAutor*. Disfrute del MNAV”. Luego de desplegar el mensaje, se le ofrece al usuario navegar por una dirección web o URL asociado al mensaje correspondiente que tiene alojadas las obras del autor en cuestión. Esto se logra luego de realizar una búsqueda en una base de datos precargada.

En este mismo sentido, se tomó la experiencia adquirida al realizar esta aplicación para realizar una segunda aplicación que permite al usuario navegar dentro de una lista de artistas cuyos cuadros se encuentran en el MNAV. Al seleccionar a uno de ellos, se pueden ver sus obras, y si se escoge una de ellas, el usuario puede acceder a distintos datos de interés como por ejemplo su nombre, su fecha de creación (si es que esta se conoce) y quizá hasta una descripción básica. Se podría en este momento, unir ambas aplicaciones en una sola, lo que generaría la posibilidad de realizar una navegación por el museo, conociendo la ubicación de cada uno de los cuadros y dando al usuario información acerca de cada uno de ellos.

Cabe destacar que se consultó con el Museo Nacional de Artes Visuales el hecho de poner leños códigos al costado de las obras, o en determinados puntos estratégicos del museo, y su respuesta fue positiva.

CAPÍTULO 3

Realidad aumentada

Definición de realidad aumentada, ejemplos. Breve mención de las cosas que se usan, modelo cámara pinhole, calibración, extracción de características (LSD, ORT), marcador utilizado. Algoritmos de estimación de pose. Algoritmos de correspondencias. Toda la rama Posit. En este capítulo se presentan las técnicas realizadas para implementar la realidad aumentada.

3.1. Definición

Es posible definir la realidad aumentada (AR del inglés *Augmented Reality*) como una vista directa o indirecta en tiempo real, de algún elemento o escena física del mundo real, a la que se le agrega información de manera virtual o digital mediante el uso de herramientas computacionales [2]. Cuando se genera una escena por medio de la realidad aumentada, conviven en ella elementos reales con elementos virtuales que buscan verse tan reales como se pueda. Es básicamente un juego de percepciones.

La realidad aumentada es un área que se encuentra en pleno desarrollo y todo el tiempo aparecen ideas novedosas y muy interesantes, lo que la hace por demás apasionante.

3.2. Fundamento

El proceso mediante el cual se logra la realidad aumentada puede dividirse básicamente en dos grandes partes. La primera es el seguimiento de la cámara. Esto es el lograr ubicar la cámara perfectamente (rotación y traslación), respecto de un eje de coordenadas arbitrario asignado al mundo real. Este seguimiento puede lograrse mediante el reconocimiento de características de las imágenes tomadas por la cámara o incluso mediante la detección de bordes o esquinas de algún marcador en particular correctamente modelado respecto del *eje del mundo* previamente mencionado. Tener un correcto modelo de los objetos del mundo respecto de este eje resulta una cuestión realmente importante para poder luego reconstruir la imagen con la información agregada digitalmente.

Una vez conocida la posición de la cámara en el mundo, resta agregar la información a la imagen. Esto es posible ya que al tener un correcto modelo del mundo respecto de su eje de coordenadas, es posible ubicar (en el modelo teórico) un objeto digital y resolviendo un sencillo sistema de ecuaciones se obtiene su posición en la imagen cuadro a cuadro.

El proceso mediante el cual se genera una imagen 2D a partir de un modelo 3D se denomina Renderización y se analizará con algo más de detalle en secciones subsiguientes. También se analizarán más a fondo el concepto del modelado 3D y el de modelado de la cámara en cuestión, ya que no todas las cámaras se comportan de igual manera desde que su construcción no es idéntica a la

construcción de ninguna otra cámara (aunque bajo ciertas hipótesis, se deberá suponer que cierto conjunto de cámaras cuenta con características similares). Para realizar entonces la transformación desde el mundo 2D al mundo 3D y viceversa, resulta de vital importancia contar con un modelo detallado y preciso de la cámara utilizada.

3.3. Modelado 3D

?Aca que puede ir?

3.4. Rendering

Esta seccion es importante!

CAPÍTULO 4

Estimación de pose monocular

Se le llama “estimación de pose” al proceso mediante el cual se calcula en qué punto del mundo y con qué orientación se encuentra determinado objeto respecto de un eje de coordenadas previamente definido al que se lo llama *ejes del mundo*. Las aplicaciones de realidad aumentada requieren de un modelado preciso del entorno respecto de estos ejes, para poder ubicar correctamente los agregados virtuales dentro del modelo y luego renderizarlos en la imagen vista por el usuario. El objeto cuya estimación de pose resulta de mayor importancia es la cámara, ya que por ésta es por donde se mira la escena y es respecto de ésta que los objetos virtuales deben ubicarse de manera consistente. Una forma de estimar la pose de la cámara es mediante el uso de las imágenes capturadas por ella misma. Asimismo, el concepto “monocular” hace referencia al uso de una sola cámara, ya que es posible trabajar con más de una.

Para poder estimar la pose de una cámara, resulta necesario modelarla adecuadamente ya que no todas las cámaras son iguales. El modelo más comunmente utilizado es el denominado *pin-hole*. Para modelar completamente la cámara se deben estimar ciertos *parámetros intrínsecos* a ésta, y eso se logra luego de realizados algunos experimentos. A la estimación de estos parámetros se le denomina *calibración de cámara*.

4.1. Calibración de cámara: modelo pin-hole [1]

4.1.1. Fundamentos y definiciones

Este modelo consiste en un centro óptico C , en donde convergen todos los rayos de la proyección y un plano imagen en el cual la imagen es proyectada. Se define “distancia focal” (f) como la distancia entre el centro óptico C y el cruce del eje óptico por el plano imagen (punto P). Ver imagen 4.1.

Para modelar el proceso de proyección (proceso en el que se asocia al punto \mathbf{M} del mundo, un punto \mathbf{m} en la imagen), es necesario referirse a varias transformaciones y varios ejes de coordenadas.

- *Coordenadas del mundo*: son las coordenadas que describen la posición 3D del punto \mathbf{M} . Se definen respecto de los *ejes del mundo* (X_m, Y_m, Z_m) . La elección de los ejes del mundo es arbitraria.
- *Coordenadas de la cámara*: son las coordenadas que describen la posición del punto \mathbf{M} respecto de los ejes de la cámara (X, Y, Z) .
- *Coordenadas de la imagen*: son las coordenadas que describen la posición del punto 2D, \mathbf{m} , respecto del centro del plano imagen, P . Los ejes de este sistema de coordenadas son (u, v) .

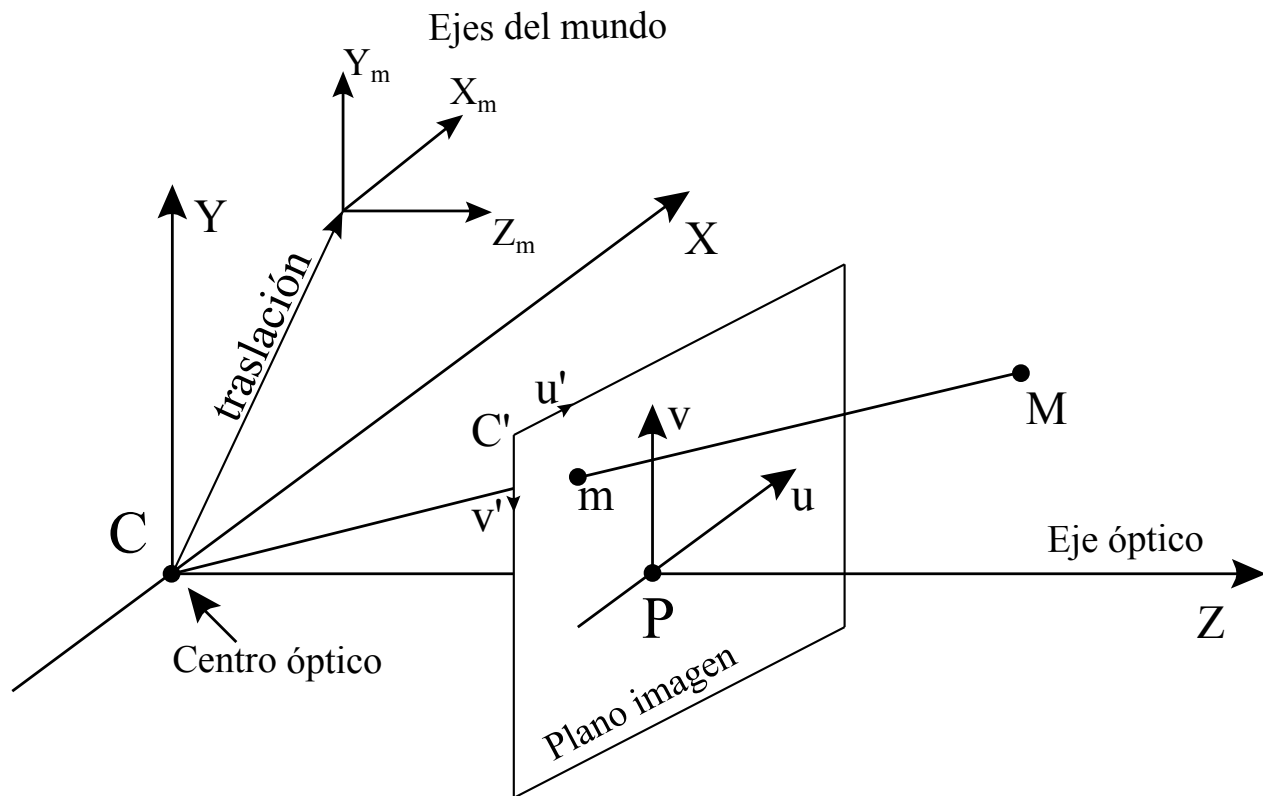


Figura 4.1: Modelo de cámara pin-hole.

- *Coordenadas normalizadas de la imagen*: son las coordenadas que describen la posición del punto 2D, \mathbf{m} , respecto del eje de coordenadas (u', v') situado en la esquina superior izquierda del plano imagen.

La transformación que lleva del punto \mathbf{M} , expresado respecto de las coordenadas del mundo, al punto \mathbf{m} , expresado respecto del sistema de coordenadas normalizadas de la imagen, se puede ver como la composición de dos transformaciones menores. La primera, es la que realiza la proyección que transforma a un punto definido respecto del sistema de coordenadas de la cámara (X, Y, Z) en otro punto sobre el plano imagen expresado respecto del sistema de coordenadas normalizadas de la imagen (u', v') . Véase que una vez calculada esta transformación, es una constante característica de cada cámara. Se le llama al conjunto de valores que definen esta transformación *parámetros intrínsecos* de la cámara. La segunda, es la transformación que lleva de expresar a un punto respecto de los ejes del mundo (X_m, Y_m, Z_m) , a los ejes de la cámara (X, Y, Z) . Esta última transformación varía conforme se mueve la cámara (respecto de los ejes del mundo) y el conjunto de valores que la definen es denominado *parámetros extrínsecos* de la cámara. Del cálculo de estos parámetros es que se obtiene la estimación de la pose de la cámara.

De lo anterior se concluye rápidamente que si se le llama PY a la matriz proyección total, tal que:

$$\mathbf{m} = PY.M,$$

entonces:

$$PY = I.E$$

donde I corresponde a la matriz proyección asociada a los parámetros intrínsecos y E corresponde a la matriz asociada a los parámetros extrínsecos.

- **Parámetros extrínsecos:** pose de la cámara.
 - Traslación: ubicación del centro óptico de la cámara respecto de los ejes del mundo.
 - Rotación: rotación del sistema de coordenadas de la cámara (X, Y, Z) , respecto de los ejes del mundo.
- **Parámetros intrínsecos:** parámetros propios de la cámara. Dependen de su geometría interna y de su óptica.
 - Punto principal ($P = [u'_p, v'_p]$): es el punto intersección entre el eje óptico y el plano imagen. Las coordenadas de este punto vienen dadas en píxeles y son expresadas respecto del sistema normalizado de la imagen.
 - Factores de conversión píxel-milímetros (d_u, d_v): indican el número de píxeles por milímetro que usa la cámara en las direcciones u y v respectivamente.
 - Distancia focal (f): distancia entre el centro óptico (C) y el punto principal (P). Su unidad es el milímetro.
 - Factor de proporción (s): indica la proporción entre las dimensiones horizontal y vertical de un píxel.

4.1.2. Matriz de proyección

En la sección anterior se vió que es posible hallar una “matriz de proyección” PY que dependa tanto de los parámetros intrínsecos de la cámara como de sus parámetros extrínsecos:

$$m = PY.M$$

donde M y m son los puntos ya definidos y vienen expresados en *coordenadas homogéneas*. Por más información acerca de este tipo de coordenadas ver [8].

Para determinar la forma de la matriz de proyección se estudia cómo se relacionan las coordenadas de M con las coordenadas de m ; para hallar esta relación se debe analizar cada transformación, entre los sistemas de coordenadas mencionados con anterioridad, por separado.

- **Proyección 3D - 2D:** de las coordenadas homogéneas del punto M expresadas en el sistema de coordenadas de la cámara (X_0, Y_0, Z_0, T_0) , a las coordenadas homogéneas del punto m expresadas en el sistema de coordenadas de la imagen (u_0, v_0, s_0) :
Se desprende de la imagen 4.1 y algo de geometría proyectiva la siguiente relación entre las coordenadas y la distancia focal (f):

$$\frac{f}{Z_0} = \frac{u_0}{X_0} = \frac{v_0}{Y_0}$$

A partir de la relación anterior:

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \frac{f}{Z_0} \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

Expresado en forma matricial, en coordenadas homogéneas:

$$\begin{pmatrix} u_0 \\ v_0 \\ s_0 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix}$$

- **Transformación cámara imagen:** de las coordenadas homogéneas del punto \mathbf{m} expresadas respecto del sistema de coordenadas de la imagen (u_0, v_0, s_0) , a las coordenadas homogéneas de él mismo pero expresadas respecto del sistema de coordenadas normalizadas de la imagen (u'_0, v'_0, s'_0) :

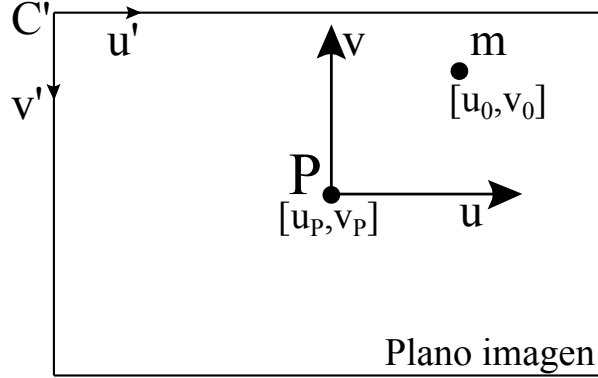


Figura 4.2: Relación entre el sistema de coordenadas de la imagen y el sistema de coordenadas normalizadas de la imagen.

Se les suma, a las coordenadas de \mathbf{m} respecto del sistema de la imagen, la posición del punto P respecto del sistema normalizado de la imagen (u'_p, v'_p) . Las coordenadas de \mathbf{m} dejan de expresarse en milímetros para expresarse en píxeles. Aparecen los factores de conversión d_u y d_v :

$$\begin{aligned} u'_0 &= d_u \cdot u_0 + u'_p \\ v'_0 &= d_v \cdot v_0 + v'_p \end{aligned}$$

Se obtiene entonces la siguiente relación matricial, en coordenadas homogéneas:

$$\begin{pmatrix} u'_0 \\ v'_0 \\ s'_0 \end{pmatrix} = \begin{pmatrix} d_u & 0 & u'_p \\ 0 & d_v & v'_p \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ v_0 \\ 1 \end{pmatrix}$$

- **Matriz de parámetros intrínsecos (I):** de las coordenadas homogéneas del punto \mathbf{M} expresadas en el sistema de coordenadas de la cámara (X_0, Y_0, Z_0, T_0) , a las coordenadas homogéneas del punto \mathbf{m} expresadas respecto del sistema de coordenadas normalizadas de la imagen (u'_0, v'_0, s'_0) :

Se obtiene combinando las dos últimas transformaciones. Nótese que como ya se aclaró, depende únicamente de parámetros propios de la construcción de la cámara:

$$I = \begin{pmatrix} d_u \cdot f & 0 & u'_p & 0 \\ 0 & d_v \cdot f & v'_p & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- **Matriz de parámetros extrínsecos (E):** de las coordenadas homogéneas del punto \mathbf{M} expresadas respecto del sistema de coordenadas del mundo $(X_{m0}, Y_{m0}, Z_{m0}, T_{m0})$, a las coordenadas homogéneas de él mismo pero expresadas respecto del sistema de coordenadas de la cámara (X_0, Y_0, Z_0, T_0) :

Se obtiene de estimar la pose de la cámara respecto de los ejes del mundo y es la combinación

de, primero una rotación R , y luego una traslación T . Se obtiene entonces la siguiente representación matricial:

$$\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ T_0 \end{pmatrix} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_{m0} \\ Y_{m0} \\ Z_{m0} \\ T_{m0} \end{pmatrix}$$

donde la matriz de parámetros extrínsecos desarrollada toma la forma:

$$E = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Matriz de proyección (PY):** de las coordenadas homogéneas del punto \mathbf{M} expresadas respecto del sistema de coordenadas del mundo $(X_{m0}, Y_{m0}, Z_{m0}, T_{m0})$, a las coordenadas homogéneas del punto \mathbf{m} expresadas respecto del sistema de coordenadas normalizadas de la imagen (u'_0, v'_0, s'_0) :

Es la proyección total y se obtiene combinando las dos transformaciones anteriores:

$$\begin{pmatrix} u'_0 \\ v'_0 \\ s'_0 \end{pmatrix} = \begin{pmatrix} d_u \cdot f & 0 & u'_p & 0 \\ 0 & d_v \cdot f & v'_p & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_{m0} \\ Y_{m0} \\ Z_{m0} \\ T_{m0} \end{pmatrix}$$

Nota: Este modelo no tiene en cuenta los efectos de distorsión de la lente.

4.2. Algoritmos de extracción de características

4.2.1. Segmentos

4.3. Marcadores

La inclusión de *marcadores*, *marcas de referencia* o *fiduciales*, en inglés *markers*, *landmarks* o *fiducials*, en la escena ayuda al problema de extracción de características y por lo tanto al problema de estimación de pose [9]. Estos por construcción son elementos que presentan una detección estable en la imagen para el tipo de característica que se desea extraer así como medidas fácilmente utilizables para la estimación de la pose.

Se distinguen dos tipos de *fiduciales*. El primer tipo son los que se llaman puntos *fiduciales* por que proveen una correspondencia de puntos entre la escena y la imagen. El segundo tipo, *fiduciales planares*, se pueden obtener mediante la construcción en una geometría coplanar de una serie de *puntos fiduciales* identificables como esquinas. Un único *fiducial planar* puede contener por si solo todas las seis restricciones espaciales necesarias para definir el marco de coordenadas.

Como se explica en la sección 4.4 el problema de estimación de pose requiere de una serie de correspondencias $\mathbf{M}_i \leftrightarrow \mathbf{m}_i$ entre el puntos 3D en la escena en coordenadas del mundo y puntos en la imagen. El enfoque elegido

4.3.1. Marcador QR

El enfoque inicial elegido para la detección de *puntos fiduciales* para marcadores parte del trabajo de fin de curso de Matías Tailanian para el curso *Tratamiento de imágenes por computadora* de Facultad de Ingeniería, Universidad de la Republica¹. La elección se basa principalmente en los buenos resultados obtenidos para dicho trabajo con un enfoque relativamente simple. El trabajo desarrolla, entre otras cosas, un diseño de marcador y un sistema de detección de marcadores basado en el detector de segmentos LSD[7] por su buena performance y aparente bajo costo computacional.

El marcador utilizado está basado en la estructura de detección incluida en los códigos *QR* y se muestra en la figura 4.3. Éste consiste en tres grupos idénticos de tres cuadrados concéntricos superpuestos de tal forma que los lados de cada uno de tres cuadrados son visualizables. A diferencia de los códigos *QR* la disposición de los grupos de cuadrados se distintos para evitar ambigüedades en la determinación de su posicionamiento espacial. Estas dos características son esenciales para la extracción de los *puntos fiduciales* de forma coherente, es decir, las correspondencias tienen que poder ser determinadas completamente bajo criterios razonables.

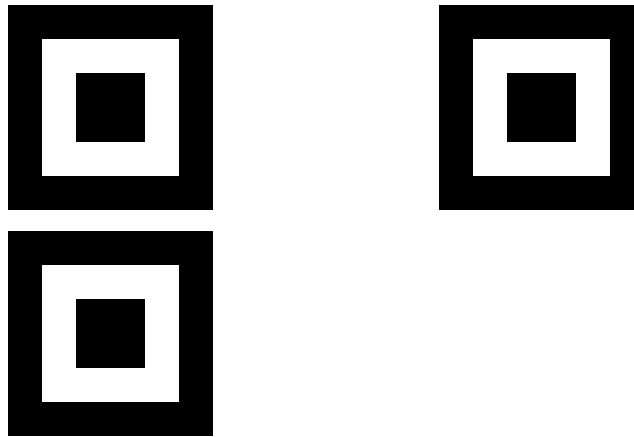


Figura 4.3: Marcador propuesto basado en la estructura de detección de códigos QR.

4.3.1.1. Elementos estructura del marcador

En la presente subsección se presentan algunas definiciones de las estructuras formantes del marcador. Estas serán de utilidad para el diseño y formarán un flujo natural y escalable para el desarrollo del algoritmo de determinación de correspondencias.

Los elementos más básicos en la estructura son los *segmentos* los cuales consisten en un par de puntos en la imagen, $\mathbf{p} = (p_x, p_y)$ y $\mathbf{q} = (q_x, q_y)$. Estos segmentos serán los lados del cuadrilátero, el próximo elemento en la estructura del marcador.

Un *cuadrilátero* o $Q1$ está determinado por cuatro segmentos conexos y distintos entre sí. La propiedad de conexión se relaja a la intersección de dos discos de un cierto radio en torno a los puntos de dos segmentos vecinos. El cuadrilátero tiene dos propiedades notables; el *centro* definido como el punto medio de sus vértices y el *perímetro* definido como la suma de sus cuatro lados. Los *vértices* de un cuadrilátero se determinan mediante la intersección, en un sentido amplio, de dos segmentos contiguos.

Un *grupo de cuadriláteros* o $Q1Set$, se construye a partir de M cuadriláteros, de distinto tamaño, que comparten un mismo centro, con $M > 1$. A partir de dichos cuadriláteros se construye una lista ordenada $(Q1[0], Q1[1], \dots, Q1[M-1])$ en donde el orden viene dado por el valor de perímetro de

¹Autoposicionamiento 3D - <http://sites.google.com/site/autoposicionamiento3d/>

cada Q1. Se define el *centro del grupo de cuadriláteros* como el baricentro, o promedio ponderado, de los centros de cada Q1 de la lista ordenada.

Finalmente el *marcador QR* estará constituido por N *grupos de cuadriláteros* dispuestos en una geometría particular. Esta geometría debe determinar un sistema de coordenadas, un origen y dos ejes. Se tendrá una lista ordenada $(Q1Set[0], Q1Set[1], \dots, Q1Set[N-1])$ en donde el orden se determinará mediante la geometría de los mismos.

Un marcador proveerá un numero de $4 \times M \times N$ vertices y por lo tanto la misma cantidad de puntos fiduciales para proveer las correspondencias $\mathbf{M}_i \leftrightarrow \mathbf{m}_i$ para el algoritmo de estimación de pose.

4.3.1.2. Diseño

Un detalle del marcador se muestra en la figura 4.4 en donde se define el grupo i de cuadrados concentricos como el $Q1Set[i]$ (sets de cuadriláteros) y se definen los respectivos centros \mathbf{c}_i para cada $Q1Set[i]$. Se considera además un eje de coordenadas que queda definido los vectores normalizados.

$$\mathbf{x} = \frac{\mathbf{c}_1 - \mathbf{c}_0}{\|\mathbf{c}_1 - \mathbf{c}_0\|}$$

$$\mathbf{y} = \frac{\mathbf{c}_2 - \mathbf{c}_0}{\|\mathbf{c}_2 - \mathbf{c}_0\|}$$

Por otro lado la disposición de los $Q1Set$ es tal que la distancia indicada \mathbf{d}_{01} definida como la norma del vector entre los centros \mathbf{c}_1 y \mathbf{c}_0 es significativamente mayor que la distancia \mathbf{d}_{02} definida como la norma del vector entre los centros \mathbf{c}_2 y \mathbf{c}_1 .

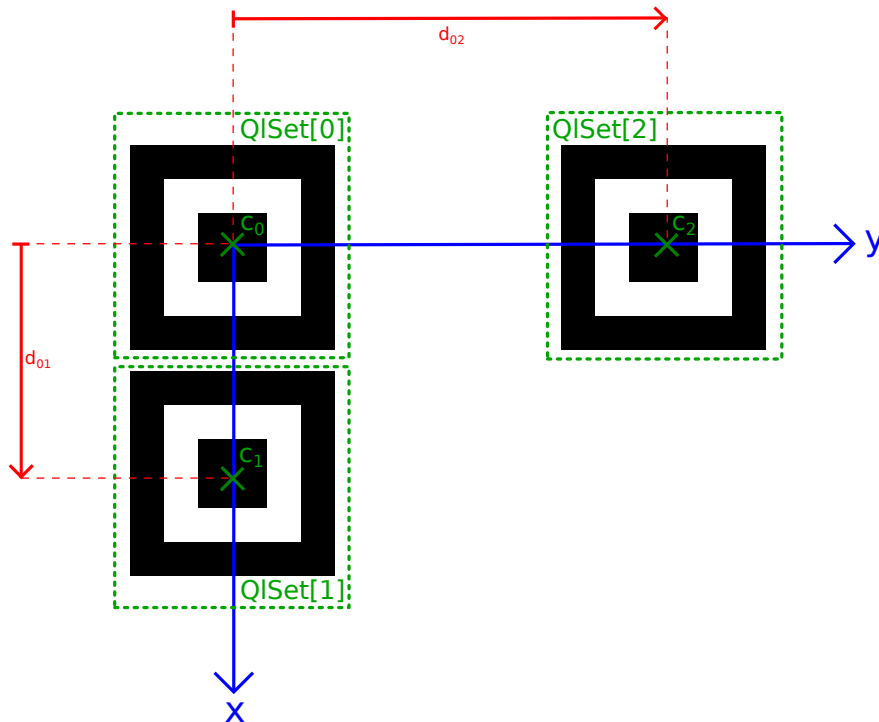


Figura 4.4: Detalle del marcador propuesto formando un sistema de coordenadas.

En base al sistema de coordenadas definido en la figura 4.4 se puede fijar un orden determinado para los *puntos fiduciales* que serán utilizados como correspondencias entre la imagen y la escena. Éstos se toman partiendo del cuadrado

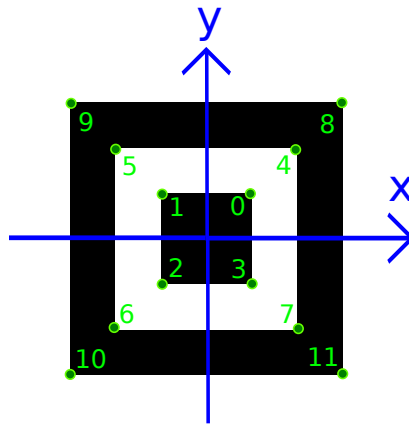


Figura 4.5: Detalle de un QISet indicando el orden de los puntos basados en el eje de coordenadas definido previamente.

4.3.1.3. Filtrado de segmentos

4.3.1.4. Determinación de correspondencias

4.4. Algoritmos de estimación de pose monocular

La estimación de pose consiste en calcular la posición y la orientación de un objeto a partir de una imagen del mismo. Dentro las técnicas de estimación de pose, se encuentran aquellas en las que se debe conocer la correspondencia entre puntos de la imagen y puntos del modelo, y aquella en las que la correspondencia se resuelve en conjunto con la estimación de pose.

conociendo la correspondencia entre puntos característicos en la imagen y puntos del objeto. Mediante los algoritmos de extracción de características vistos anteriormente se obtienen puntos claves del objeto los

- POSIT
- BABAB
- ABABA
- BABAB

4.5. POSIT

El algoritmo Posit permite calcular la pose de un objeto conocido a partir de una imagen en la que se conocen puntos característicos del objeto. Las correspondencias entre los puntos detectados en la imagen y los puntos del objeto deben ser conocidas y deben ser más de cuatro. En la versión original del algoritmo, presentado en (ref), los puntos del objeto utilizados para estimar la pose no deben ser coplanares. Debido a que el marcador utilizado es plano fue necesario utilizar una versión del algoritmo que soporta puntos del modelo coplanares (ref).

4.5.1. Hipotesis de trabajo

- Modelo de cámara pinhole. El centro de la cámara está en el punto O , el plano de imagen G se encuentra a distancia f de O . Los ejes O_x y O_y apuntan según las filas y columnas del sensor

de la cámara, el eje O_z apunta según el eje óptico. Los versores para estos ejes son i , j y k .

- n puntos $M_0, M_1, \dots, M_i, \dots, M_{n-1}$ del objeto, ubicados en el campo de vista de la cámara (fov).
- Se toma M_0 como punto de referencia del objeto, el eje de coordenadas respecto al objeto queda definido por (u, v, w) . Como la geometría del objeto se asume conocida, también se conocen las coordenadas (U_i, V_i, W_i) de los puntos M_i en el eje de coordenadas del objeto
- Las imágenes de los puntos M_i son los puntos m_i con coordenadas (x_i, y_i) en el plano imagen también conocidas.
- Las coordenadas (X_i, Y_i, Z_i) de los puntos M_i en el eje de coordenadas de la cámara son desconocidas.

4.5.2. Problema a resolver

Se busca computar la matriz de rotación y el vector de traslación del objeto. La matriz de rotación \mathbf{R} del objeto, es la matriz cuyas filas son las coordenadas de los versores i , j y k del sistema de coordenadas de la cámara en el sistema de coordenadas del objeto (u, v, w) .

La matriz \mathbf{R} queda:

$$R = \begin{pmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{pmatrix}$$

Para obtener la matriz de rotación solo es necesario obtener los versores \mathbf{i} y \mathbf{j} , el versor \mathbf{k} se obtiene de realizar el producto vectorial $\mathbf{i} \times \mathbf{j}$. El vector de traslación es el vector que va del centro del objeto M_0 a el centro del sistema de coordenadas de la cámara O . Por lo tanto las coordenadas del vector de traslación son (X_0, Y_0, Z_0) . Si este punto M_0 es uno de los puntos visibles en la imagen, entonces el vector \mathbf{T} esta alineado con el vector Om_0 y es igual a $(Z_0/f)Om_0$.

Por lo tanto la pose queda determinada si se conocen \mathbf{i} , \mathbf{j} y Z_0 .

4.5.3. Proyeccion ortografica escalada (SOP)

La SOP es una aproximacion a la proyeccion perspetitva. Dado cualquier punto del objeto M_i con coordenadas (X_i, Y_i, Z_i) , se asume que todas las profundidades Z_i son similares entre si y pueden ser aproximadas por Z_0 que corresponde a la profundidad del punto de referencia M_0 . La SOP de un punto M_i es un punto p_i en el plano G con coordenadas (x'_i, y'_i) con

$$x'_i = fX_i/Z_0 \quad y'_i = fY_i/Z_0$$

mientras que para una proyección perspectiva lo que se obtiene es el punto m_i de coordenadas (x_i, y_i) con

$$x_i = fX_i/Z_i \quad y_i = fY_i/Z_i$$

El factor $s = f/Z_0$ es el factor de escala que pone la S en SOP. Una vez que se realiza la proyección ortogonal del punto M_i sobre el plano paralelo al plano imagen que contiene al punto M_0 que da lugar al punto P_i , se realiza la proyección perspectiva de este punto sobre el plano imagen. Analíticamente hacer esta proyeccion equivale a multiplicar las coordenadas del punto P_i de coordenadas (X_i, Y_i, Z_0) por el factor de escala s . Las coordenadas de la SOP tambien se pueden expresar como

$$x'_i = fX_0/Z_0 + f(X_i - X_0)/Z_0 = x_0 + s(X_i - X_0)$$
$$y'_i = y_0 + s(Y_i - Y_0)$$

En la figura ?? se puede ver la construccion geometrica de la SOP.

4.5.4. Ecuaciones fundamentales

Una vez obtenida la SOP de los puntos del modelo

Este algoritmo utiliza el algoritmo llamado POS(Pose from Orthography and Scaling), en el que se aproxima la pose por la pose obtenida a partir de una imagen SOP(Scale Orthographic Projection). Una vez obtenida la pose aproximada se vuelve crear una imagen SOP y se estima una nueva pose. Este procedimiento es repetido hasta que la pose estimada sea buena.

4.6. validación de algoritmos

- ABABA
- BABAB
- ABABA
- BABAB

[4]

Bibliografía

- [1] J. García Ocón. Autocalibración y sincronización de múltiples cámaras plz. 2007.
- [2] B. Furht. *The Handbook of Augmented Reality*. 2011.
- [3] C. Avellone and G. Capdehourat. Posicionamiento indoor con señales wifi. 2010.
- [4] Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. Simultaneous pose and correspondence determination using line features. pages 424–431, 2003.
- [5] Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. Softposit: Simultaneous pose and correspondence determination. pages 424–431, 2002.
- [6] Daniel F. DeMenthon and Larry S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.
- [7] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, April 2010.
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [9] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.