
Índice general

Índice general	1
1. Detección	2
1.1. Primitivas	2
1.1.1. Puntos y esquinas	2
1.1.1.1. Algoritmos de puntos y esquinas	2
1.1.2. Líneas	2
1.1.2.1. Algoritmos de líneas	2
1.1.3. Segmentos	2
1.1.3.1. Segmentos	2
1.1.4. Cilindros y esferas	2
1.1.4.1. Algoritmos de cilindros y esferas	2
1.2. Detección sin primitivas markerless	2
1.3. Marcadores	2
1.3.1. Marcador QR	3
1.3.1.1. Diseño: estructura del marcador	3
1.3.1.2. Diseño	4
1.3.1.3. Filtrado de segmentos	4
1.3.1.4. Determinación de correspondencias	4
2. LSD: “Line Segment Detection”	6
2.1. Introducción	6
2.2. <i>Line-support regions</i>	6
2.3. Aproximación de las regiones por rectángulos	7
2.4. Validación de segmentos	8

CAPÍTULO 1

Detección

Hola

1.1. Primitivas

1.1.1. Puntos y esquinas

1.1.1.1. Algoritmos de puntos y esquinas

1.1.2. Líneas

1.1.2.1. Algoritmos de líneas

1.1.3. Segmentos

1.1.3.1. Segmentos

1.1.4. Cilindros y esferas

1.1.4.1. Algoritmos de cilindros y esferas

1.2. Detección sin primitivas markerless

SIFT (puntero a capítulo que tiene SIFT para reconocimiento) SURF, ETC ETC.

1.3. Marcadores

Marcadores que se usan. Limitaciones

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 [?]. 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 ?? 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

1.3.1. 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[?] 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 1.1. 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.

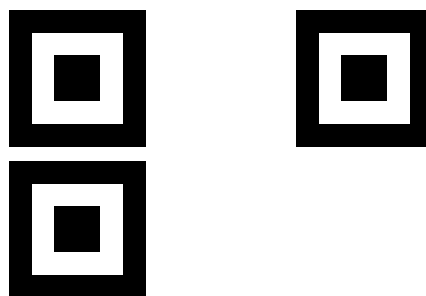


Figura 1.1: Marcador propuesto basado en la estructura de detección de códigos QR.

1.3.1.1. Diseño: 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 formaran un flujo natural y escalable para el desarrollo del algoritmo de determinación de correspondencias.

Los elementos mas 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.

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

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 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 número de $4 \times M \times N$ vértices 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.

1.3.1.2. Diseño

Un detalle del marcador se muestra en la figura 1.2 en donde se define el grupo i de cuadrados concéntricos 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 por 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 .

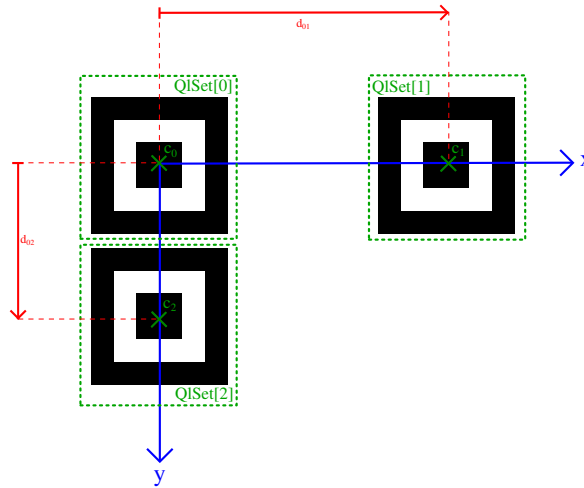


Figura 1.2: Detalle del marcador propuesto formando un sistema de coordenadas.

En base al sistema de coordenadas definido en la figura 1.2 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

1.3.1.3. Filtrado de segmentos

1.3.1.4. Determinación de correspondencias

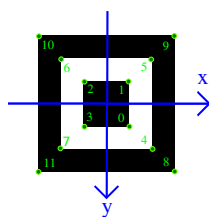


Figura 1.3: Detalle de un QISet indicando el orden de los puntos basados en el eje de coordenadas definido previamente.

CAPÍTULO 2

LSD: “Line Segment Detection”

2.1. Introducción

LSD es un algoritmo de detección de segmentos publicado por Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel y Gregory Randall en abril de 2010. Es temporalmente lineal, tiene precisión inferior a un píxel y no requiere de un tuneo previo de parámetros, como casi todos los demás algoritmos de idéntica función; puede ser considerado el estado del arte en cuanto a detección de segmentos en imágenes digitales. Como cualquier otro algoritmo de detección de segmentos, LSD basa su estudio en la búsqueda de contornos angostos dentro de la imagen. Estos son regiones en donde el nivel de brillo de la imagen cambia notoriamente entre píxeles vecinos, por lo que el gradiente de la misma resulta de vital importancia. Se genera previo al análisis de la imagen, un campo de orientaciones asociadas a cada uno de los píxeles denominado por los autores *level-line orientation field*. Dicho campo se obtiene de calcular las orientaciones ortogonales a los ángulos asociados al gradiente de la imagen. Luego, LSD puede verse como una composición de tres pasos:

- (1) División de la imagen en las llamadas *line-support regions*, que son grupos conexos de píxeles con idéntica orientación, hasta cierta tolerancia.
- (2) Búsqueda del segmento que mejor aproxime cada *line-support region*: aproximación de las regiones por rectángulos.
- (3) Validación o no de cada segmento detectado en el punto anterior.

Los puntos (1) y (2) están basados en el algoritmo de detección de segmentos de Burns, Hanson y Riseman y el punto (3) es una adaptación del método *a contrario* de Desolneux, Moisan y Morel.

2.2. *Line-support regions*

El primer paso de LSD es el dividir la imagen en regiones conexas de píxeles con igual orientación, a menos de cierta tolerancia τ , llamadas *line-support regions*. El método para realizar tal división es del tipo “región creciente”; cada región comienza por un píxel y cierto ángulo asociado al mismo. Luego, se testean sus ocho vecinos y los que cuenten con un ángulo similar al del píxel original son incluidos en la región. En cada iteración el ángulo asociado a la región es calculado como el promedio de las orientaciones de cada píxel dentro de la *line-support region*; la iteración

termina cuando ya no se pueden agregar más píxeles a la misma.

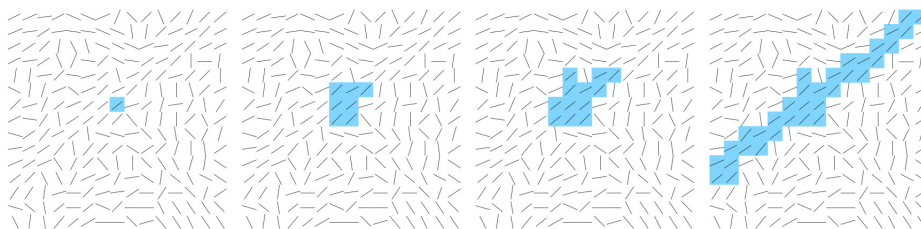


Figura 2.1: Proceso de crecimiento de una región. El ángulo asociado cada píxel de la imagen está representado por los pequeños segmentos y los píxeles coloreados representan la formación de la región. Fuente [?].

Los píxeles agregados a una región son marcados de manera que no se vuelvan a testear. Para mejorar el desempeño del algoritmo, las regiones comienzan a evaluarse por los píxeles con gradientes de mayor amplitud ya que estos representan mejor los bordes.

Existen algunos casos puntuales en los que el proceso de búsqueda de *line-support regions* puede arrojar errores. Por ejemplo, cuando se tienen dos segmentos que se juntan y que son colineales a no ser por la tolerancia τ descrita anteriormente, se detectarán ambos segmentos como uno solo; ver figura 2.2. Este potencial problema es heredado del algoritmo de Burns, Hanson y Riseman.



Figura 2.2: Potencial problema heredado del algoritmo de Burns, Hanson y Riseman. Izq.: Imagen original. Ctro.: Segmento detectado. Der.: Segmentos que deberían haberse detectado. Fuente [?].

Sin embargo, LSD plantea un método para ahorrarse este tipo de problemas. Durante el proceso de crecimiento de las regiones, también se realiza la aproximación rectangular a dicha región (paso (2) de los tres definidos anteriormente); y si menos del 50 % de los píxeles dentro del rectángulo corresponden a la *line-support region*, entonces lo que se tiene no es un segmento. Se detiene entonces el crecimiento de la región.

2.3. Aproximación de las regiones por rectángulos

Cada *line-support region* debe ser asociada a un segmento. Cada segmento será determinado por su centro, su dirección, su anchura y su longitud. A diferencia de lo que pudiese dictar la intuición, la dirección asociada al segmento no se corresponde con la asociada a la región (el promedio de las direcciones de cada uno de los píxeles). Sin embargo, se elige el centro del segmento como el centro de masa de la región y su dirección como el eje de inercia principal de la misma; la magnitud del gradiente asociado a cada píxel hace las veces de masa. La idea detrás de este método es que los píxeles con un gradiente mayor en módulo, se corresponden mejor con la percepción de un borde. La anchura y la longitud del segmento son elegidos de manera de cubrir el 99 % de la masa de la región.

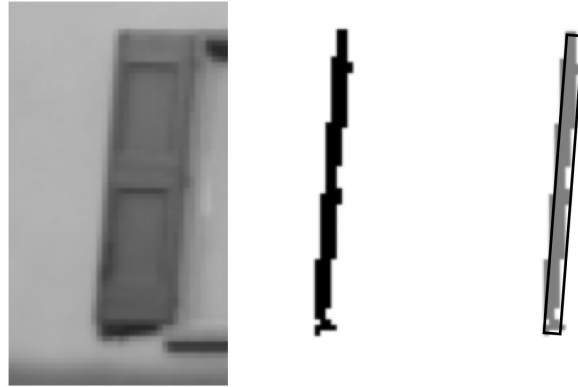


Figura 2.3: Búsqueda del segmento que mejor aproxime cada *line-support region*: aproximación de una región por un rectángulo. Izq.: Imagen original. Ctro.: Una de las regiones computadas. Der.: Aproximación rectangular que cubre el 99 % de la masa de la región. Fuente [?].

2.4. Validación de segmentos

Wow!

[?].