CAPÍTULO 1

Detección

Hola

1.1. Típos de características

Intro y mas breves definiciones?.
Bordes, esquinas, líneas, segmentos de línea, regiones (blobs).

1.2. Bordes y esquinas

1.2.1. Detector de bordes de Canny

1.2.2. Detector de bordes y esquinas de Harris

1.2.3. SUSAN Y FAST

1.3. Líneas y segmentos de línea

1.3.1. Detector de líneas de Hough

1.3.2. Detector de segmentos de línea: LSD

1.4. Regiones y puntos de interés

1.4.1. FAST

1.4.2. Blobs

1.5. Detección sin primitivas markerless

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

1.6. 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 [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 facilmente 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 puntos 3D en la escena en coordenadas del mundo y puntos en la imagen.

1.7. 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 Republica[1]. 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 1.1. Éste consiste en trés grupos idénticos de tres cuadrados concéntricos superpuestos de tal forma que los lados de cada uno de trés cuadrados son visualizables. A diferencia de los códigos *QR* la disposición de los grupos de cuadrados es distinto para evitar ambiguedades en la determinación de su posicionamiento espacial. Estas dos características son escenciales para la extracción de los *puntos fiduciales* de forma coherente, es decir, las correspondencias tienen que poder ser determinadas completamente bajo criterios razonables.

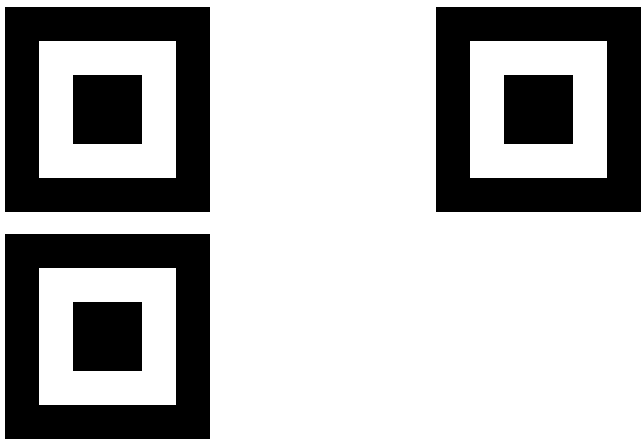

Figura 1.1: Marcador propuesto basado en la estructura de detección de códigos QR.

[1] Autoposicionamiento 3D - http://sites.google.com/site/autoposicionamiento3d/

1.7.1. Estructura del marcador

A continuación se presentan algunas definiciones de las estructuras básicas que constituyen el marcador propuesto. Estas son de utilidad para el diseño y forman 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* forman lo que son los lados del *cuadrilátero*, el próximo elemento estructural del marcador.

Un *cuadrilátero* o *quadrilateral* en inglés, al que se le denomina Ql, está determinado por cuatro segmentos conexos y distintos entre sí. El cuadrilátero tiene dos propiedades notables; el *centro* definido como el punto medio entre sus cuatro vértices y el *perímetro* definido como la suma de el largo de sus cuatro lados. Los *vértices* de un *cuadrilátero* se determinan mediante la intersección, en sentido amplio, de dos segmentos contiguos. Es decir, si s_1 es contiguo a s_2 dadas las recta r_1 que pasa por los puntos $\mathbf{p}_1$, $\mathbf{q}_1$ del segmento s_1 y la recta r_2 que pasa por los puntos $\mathbf{p}_2$, $\mathbf{q}_2$ del segmento s_2, se determina el vértice correspondiente como la intersección $r_1 \cap r_2$.

A un *conjunto de cuadriláteros* o *quadrilateral set* se le denomina QlSet, se construye a partir de M cuadriláteros, que comparten un mismo centro, y se diferencian por un factor de escala, con $\mathrm{M} > 1$. A partir de dichos cuadriláteros se construye un lista ordenada $(\mathrm{Ql}[0], \mathrm{Ql}[1], \ldots, \mathrm{Ql}[\mathrm{M}-1])$ en donde el orden viene dado por el valor de *perímetro* de cada Ql. Se define el *centro del grupo de cuadriláteros* como el promedio de los centros de cada Ql de la lista ordenada.

Finalmente el *marcador QR* está constituido por N *conjuntos de cuadriláteros* dispuestos en una geometría particular. Esta geometría permite la determinación un sistema de coordenadas; un origen y dos ejes a utilizar. Se tiene una lista ordenada $(\mathrm{QlSet}[0], \mathrm{QlSet}[1], \ldots, \mathrm{QlSet}[\mathrm{N}-1])$ en donde el orden se puede determinar mediante la geometría de los mismos o a partir de hipótesis razonables.

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

1.7.2. Diseño

En base a las estructuras previamente definidas es que se describe el diseño del marcador. Como ya se explicó se toma un marcador tipo *QR* basado en *cuadriláteros* y mas específicamente en tres conjuntos de tres cuadrados dispuestos en como se muestra en la figura 1.1.

Los tres *cuadriláteros* correspondientes a un mismo *conjunto de cuadriláteros* tienen idéntica alineación e idéntico centro. Los diferencia un factor de escala, esto es, $\mathrm{Ql}[0]$ tiene lado l mientras que $\mathrm{Ql}[1]$ y $\mathrm{Ql}[2]$ tienen lado $2l$ y $3l$ respectivamente. Esto se puede ver en la figura 1.2. Adicionalmente se define un sistema de coordenadas con centro en el centro del QlSet y ejes definidos como **x** horizontal a la derecha e **y** vertical hacia abajo. Esta convención en las direcciones de los ejes es muy utilizada en el ambiente del *tratamiento de imágenes* para definir las direcciones de los ejes de una imagen. Definido el sistema de coordenadas de puede fijar un orden a los *vértices* v_{j_1} de cada *cuadrilátero* Ql[j] como,

$$v_{j_0} = (a/2, a/2) \qquad v_{j_2} = (-a/2, -a/2)$$
$$v_{j_1} = (a/2, -a/2) \qquad v_{j_3} = (-a/2, a/2)$$

con $a = (j+1) \times l$. El orden aquí explicado se puede ver también junto con el sistema de coordenadas en la figura 1.3.

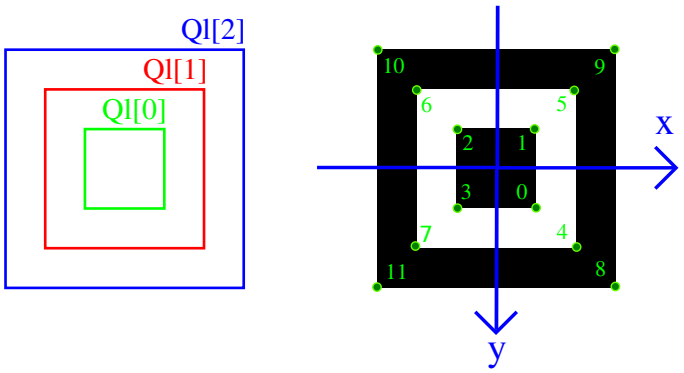

Figura 1.2: Detalle de un QlSet. Orden de los *cuadriláteros* a la izquiera. Órden de los *vértices* a la derecha.

Un detalle del marcador completo se muestra en la figura 1.3 en donde se define el conjunto i de cuadriláteros concéntricos como el QlSet[i] y se definen los respectivos centros $\mathbf{c}_i$ para cada QlSet[i]. El sistema de coordenadas del *marcador QR* tiene centro en el centro del QlSet[0] y ejes de coordenadas idénticos al definido para cada Ql. Se tiene además que los ejes de coordenadas pueden ser obtenidos mediante los vectores normalizados,

$$\mathbf{x} = \frac{\mathbf{c}_1 - \mathbf{c}_0}{||\mathbf{c}_1 - \mathbf{c}_0||} \quad \mathbf{y} = \frac{\mathbf{c}_2 - \mathbf{c}_0}{||\mathbf{c}_2 - \mathbf{c}_0||} \tag{1.1}$$

La disposición de los QlSet es tal que la distancia indicada d_{01} definida como la norma del vector entre los centros $\mathbf{c}_1$ y $\mathbf{c}_0$ es significativamente mayor que la distancia d_{02} definida como la norma del vector entre los centros $\mathbf{c}_2$ y $\mathbf{c}_1$. Esto es, $d_{01} \gg d_{02}$. Este criterio facilita la identificación de los *vértices* de los QlSet entre sí basados únicamente en la posición de sus centros y se explicará en la parte de determinación de correspondencias (sec.: 1.7.3.3).

1.7.2.1. Parámetros de diseño

Provisto el diseño del marcador antes descrito, quedan definidos ciertos parámetros **estructurales** que fueron de tomados fijos a lo largo del proyecto pero que podrían se variados en trabajos futuros asociados. Estos parámetros son:

- M: cantidad de *conjuntos de cuadriláteros*.
- N: cantidad de *cuadriláteros* por *conjuntos de cuadriláteros*.
- Geometría: geometría de los cuadriláteros (Ql).
- Disposición: disposición espacial de los *conjuntos de cuadriláteros* (QlSet).

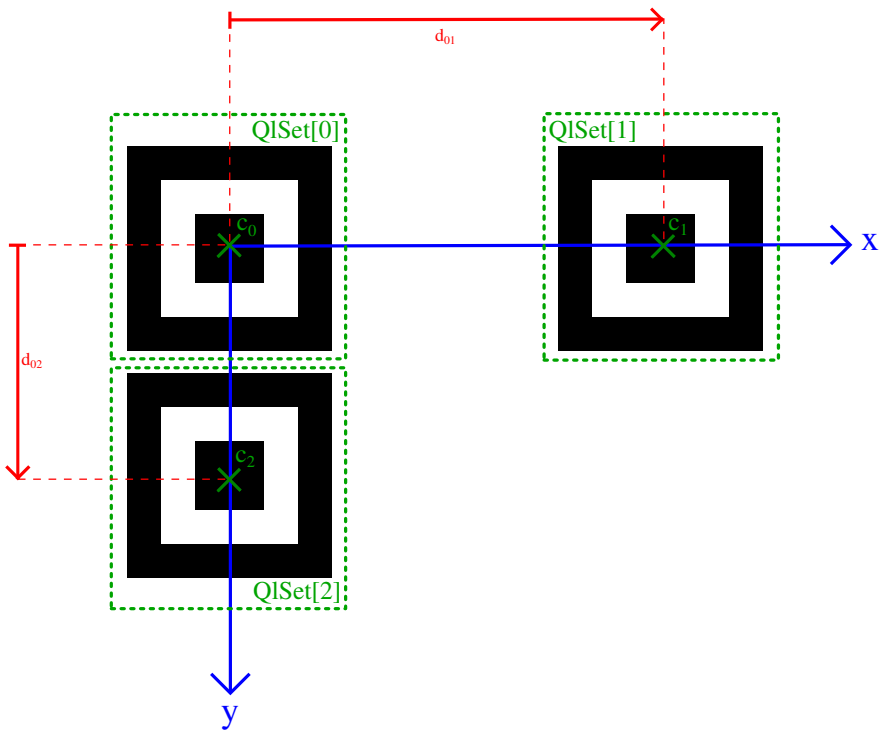

Figura 1.3: Detalle del marcador propuesto formando un sistema de coordenadas.

El criterio de elección de M y N parte del diseño los códigos QR como ya fue explicado. La detección por segmentos de línea resulta una cantidad de $3 \times$ QlSet's conteniendo $3 \times$ Ql's cada uno. Bajo esta elección de parámetros se tienen 36 *segmentos* y *vértices*. Se tienen entonces un número de puntos característicos razonable para la estimación de pose.

La elección de *cuadrados* como parámetro de geometría se basa en la necesidad de tener igual resolución en los dos ejes del marcador. De esta forma se asegura una distancia límite en donde, en un caso ideal enfrentado al marcador, la detección de segmentos de línea falla simultaneamente en los segmentos verticales como en los horitonzales. De otra forma se tendría una dirección que limita mas que la otra desaprovechando resolución.

La disposición espacial de los *conjuntos de cuadriláteros* esta en primer lugar limitada a un plano y en segundo lugar es tal que se puede definir ejes de coordenadas ortogonales mediante los centros.

Por otro lado se tiene otro juego de parámetros que concluyen con el diseño del marcador. Estos parámetros conservan la estructura intrínseca del marcador permitiendo versatilidad en la aplicación y sin la necesidad de modificación alguna de los algoritmos desarrollados. Estos son:

- d_{ij}: distancia entre los *centros* QlSet$[j]$ con QlSet$[i]$.
- l: lado del *cuadrilátero* mas pequeño (Ql$[0]$) de los QlSet.

En este caso se debe siempre cumplir la condición impuesta previamente en donde $d_{01} \gg d_{02}$. De otra forma se deberán realizar ciertas hipótesis no genéricas o se deberá aumentar ligeramente la complejidad del algoritmo para la identificación del marcador.

1.7.2.2. Diseño *test*

Durante el desarrollo de los algoritmos de detección e identificación de los *vértices* del *marcador QR* se trabajó con determinados parámetros de diseño de dimensiones apropiadas para posibilitar el traslado y las pruebas domésticas.

1.7.2.3. Diseño *Da Vinci*

1.7.2.4. Diseño *Artigas*

1.7.2.5. Diseño *Mapa*

1.7.3. Detección

La etapa de detección del marcador se puede separar en tres grandes bloques; la detección de segmentos de línea, el filtrado de segmentos y la determinación de correspondencias (figura **??**).

1.7.3.1. Detección de segmentos de línea

La detección de segmentos de línea se realiza mediante el uso del algoritmo *LSD* el cual se detalla en el capítulo **??**. En forma resumida, dicho algoritmo toma como entrada una imagen en escala de grises de tamaño $m \times n$ y devuelve una lista de segmentos en forma de pares de puntos de origen y destino.

```
int nb;
double *img;
...
out = lsd(&nb,img,m,n);
```

1.7.3.2. Filtrado de segmentos

El filtrado de segmentos consiste en la búsqueda de conjuntos de cuatro segmentos conexos en la lísta de segmentos de línea detectados por *LSD*. Los conjuntos de segmentos conexos encontrados se devuelven en una lista similar a la de *LSD*. A continuación se realiza una breve descripción del algoritmo de filtrado de segmentos implementado.

Se parte de una lista de n segmentos de línea y se recorre en busca de segmentos vecinos. La estrategia utilizada consiste en buscar, para el i-ésimo segmento $\mathbf{s}_i$, dos segmentos vecinos $\mathbf{s}_j$ y $\mathbf{s}_k$ en forma de "U" y luego buscar un último segmento $\mathbf{s}_l$ que cierre la "U". Una vez encontrado el conjunto de cuatro segmentos conexos se marcan estos segmentos como utilizados, se guardan en una lista de salida y se continúa con el segmento $i+1$ hasta recorrer los n segmentos de la lista de entrada.

testean sus dos puntos $\mathbf{p}_i$ y $\mathbf{q}_i$ con los puntos del j-ésimo segmento $\mathbf{s}_j$, $\mathbf{p}_j$ y $\mathbf{q}_j$ para $j = (i+1,\ldots,n)$.

Dos segmentos son vecinos si se cumple que la distancia euclidiana entre puntos, d_{ij}, es menor a un cierto umbral para alguna de las combinaciones $\mathbf{p}_i \leftrightarrow \mathbf{p}_j$, $\mathbf{q}_i \leftrightarrow \mathbf{q}_j$, $\mathbf{p}_i \leftrightarrow \mathbf{q}_j$ o $\mathbf{q}_i \leftrightarrow \mathbf{p}_j$. Si se cumple alguna de las correspondencias se continúa buscando un segmento que sea vecino. Por ejemplo, si se tiene la correspondencia $\mathbf{p}_i \leftrightarrow \mathbf{p}_j$ se busca el k-ésimo segmento $\mathbf{s}_k$ que cumple que la distancia euclidiana d_{ij} es menor a cierto umbral para alguna de las combinaciones $\mathbf{q}_i \leftrightarrow \mathbf{p}_k$ y $\mathbf{q}_i \leftrightarrow \mathbf{q}_k$.

La condición de vecindad se podría refinar de forma de disminuir el riesgo de tener "falsos vecinos". Esta situación se puede dar, por ejemplo, en un grupo de segmentos paralelos cercanos y de

tamaño similar o en el caso de detectar cuadrados concéntricos en los que la distancia entre sus lados es inferior al umbral. En el primer caso se puede considerar un condición de vecindad que tome en cuenta la intersección de los segmentos y la distancia de la intersección a los puntos del segmento. De todas formas ninguna de las dos situaciones se hicieron presentes en casos prácticos y la implementación de soluciones no fue necesaria además de que aumentaría levemente la complejidad del algoritmo.

Algorithm 1: filterSegments

Data: list=$\{(\mathbf{p}_0,\mathbf{q}_0),(\mathbf{p}_1,\mathbf{q}_1),\ldots,(\mathbf{p}_{n-1},\mathbf{q}_{n-1})\}$.
Result: filteredlist=$\{(\mathbf{p}_0,\mathbf{q}_0),(\mathbf{p}_1,\mathbf{q}_1),\ldots,(\mathbf{p}_{m-1},\mathbf{q}_{m-1})\}$. Lista ordenada.
begin
  for $i=(0,1,\ldots,n-1)$ **do**
    if *segmento i no marcado* **then**
      for $z \in ImPred(y) \cap Min$ **do**
        remove the arc zy from V;
        $NbSuccInS(z) \longleftarrow NbSuccInS(z)-1$;
        move z in T to the list preceding its present list;
        {i.e. If $z \in T[k]$, move z from $T[k]$ to $T[k-1]$};
  1 **while** $S \neq \emptyset$ **do**
  REM remove x from the list of T of maximal index;
  2 **while** $|S \cap ImSucc(x)| \neq |S|$ **do**
      for $y \in S - ImSucc(x)$ **do**
        { remove from V all the arcs zy : };
        for $z \in ImPred(y) \cap Min$ **do**
          remove the arc zy from V;
          $NbSuccInS(z) \longleftarrow NbSuccInS(z)-1$;
          move z in T to the list preceding its present list;
          {i.e. If $z \in T[k]$, move z from $T[k]$ to $T[k-1]$};
        $NbPredInMin(y) \longleftarrow 0$;
        $NbPredNotInMin(y) \longleftarrow 0$;
        $S \longleftarrow S-\{y\}$;
        $AppendToMin(y)$;
    $RemoveFromMin(x)$;
end

1.7.3.3. Determinación de correspondencias

Se detalla a continuación el algoritmo de determinación de correspondencias a partir de grupos de cuatro segmentos de línea conexos. Para ese algoritmo se hace uso de los elementos estructurales del marcador de forma de desarrollar un algoritmo modular, escalable y simple.

Se toma como entrada la lista de segmentos filtrados

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_0 & \mathbf{s}_1 & \ldots & \mathbf{s}_i & \mathbf{s}_{i+1} & \mathbf{s}_{i+2} & \mathbf{s}_{i+3} & \ldots & \mathbf{s}_{n-1} \end{pmatrix}^t \tag{1.2}$$

en donde cada segmento se compone de un punto inicial $\mathbf{p}_i$ y un punto final $\mathbf{q}_i$, $\mathbf{s}_i = (\mathbf{p}_i, \mathbf{q}_i)$, con n es múltiplo de cuatro. Si i también lo es, entonces el sub-conjunto

$$\mathbf{S}_i = \begin{pmatrix} \mathbf{s}_i & \mathbf{s}_{i+1} & \mathbf{s}_{i+2} & \mathbf{s}_{i+3} \end{pmatrix}^t \tag{1.3}$$

corresponde a un conjunto de cuatro segmentos del línea conexos.

Para cada sub-conjunto $\mathbf{S}_i$ se intersectan entre sí los segmentos obteniendo una lista de cuatro vértices,

$$\mathbf{V}_i = \begin{pmatrix} \mathbf{v}_i & \mathbf{v}_{i+1} & \mathbf{v}_{i+2} & \mathbf{v}_{i+3} \end{pmatrix}^t \tag{1.4}$$

Si $\mathbf{r}_i$ es la recta que pasa por los puntos $\mathbf{p}_i$ y $\mathbf{q}_i$ del segmento $\mathbf{s}_i$, la lista de vértices se obtiene como sigue,

$$\begin{aligned} \mathbf{v}_i &= \mathbf{r}_i \cap \mathbf{r}_{i+1} \\ \mathbf{v}_{i+1} &= \mathbf{r}_i \cap \mathbf{r}_{i+2} \\ \mathbf{v}_{i+2} &= \mathbf{r}_{i+3} \cap \mathbf{r}_{i+2} \\ \mathbf{v}_{i+3} &= \mathbf{r}_{i+3} \cap \mathbf{r}_{i+1} \end{aligned}$$

resultando en dos posibles configuraciones de vértices. Las dos configuraciones se muestran en la figura 1.4 en donde una de ellas tiene sentido horario y la otra antihorario.

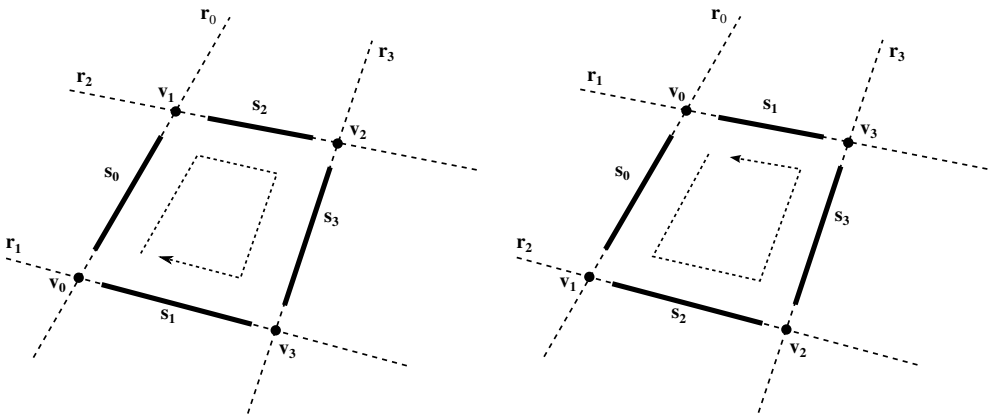

Figura 1.4: Posibles configuraciones de vértices posterior a la intersección de conjuntos de segmentos pertenecientes a un cuadrilátero.

Posterior a la intersección se realiza un chequeo sobre el valor de las coordenadas de lo vértices. Si alguno de ellos se encuentra fuera de los límites de la imagen, todo el grupo de cuatro segmentos es marcado como inválido. Este chequeo resulta en el filtado de "falsos cuadriláteros" como por ejemplo un grupo de segmentos paralelos cercanos.

Para cada uno de los conjuntos de vértices se construye con ellos un elemento *cuadrilátero* que se almacena en una lista de cuadriláteros

$$QlList = \begin{pmatrix} \mathrm{Ql}[0] & \mathrm{Ql}[1] & \dots & \mathrm{Ql}[i] & \dots & \mathrm{Ql}[\frac{n}{4}] \end{pmatrix}^t \tag{1.5}$$

A partir de esa lista de cuadriláteros, se buscan grupos de tres cuadriláteros QlSet que "compartan" un mismo centro. Para esto se recorre ordenadamente la lista en i buscando para cada cuadrilátero dos cuadriláteros j y k que cumplan que la distancia entre sus centros y el del i-ésimo cuadrilátero sea menor a cierto umbral d_{th},

$$d_{ij} = ||\mathbf{c}_i - \mathbf{c}_j|| < d_{th} \quad d_{ik} = ||\mathbf{c}_i - \mathbf{c}_k|| < d_{th} \tag{1.6}$$

Estos cuadriláteros se marcan en la lista como utilizados con ellos se forma el l-ésimo *QlSet* ordenándolos según su perímetro, de menor a mayor como $QlSet[l] = \begin{pmatrix} Ql[0] & Ql[1] & Ql[2] \end{pmatrix}$ con

$l = (0,1,2)$. Esta busqueda se realiza hasta encontrar un total de tres *QlSet* completos de forma de obtener un marcador completo, esto es, detectando todos los cuadriláteros que lo componen.

Una vez obtenida la lista $QlSetList = \begin{pmatrix} QlSet[0] & QlSet[1] & QlSet[2] \end{pmatrix}$ esta se ordena de forma que su dispoción espacial se corresponda con la del *marcador QR*. Se calculan las distancias entre los centros de cada *QlSet* y se toma el índice i como el índice que produce el vector de menor distancia,

$$\mathbf{u}_i = \mathbf{c}_{i+1} - \mathbf{c}_i.$$

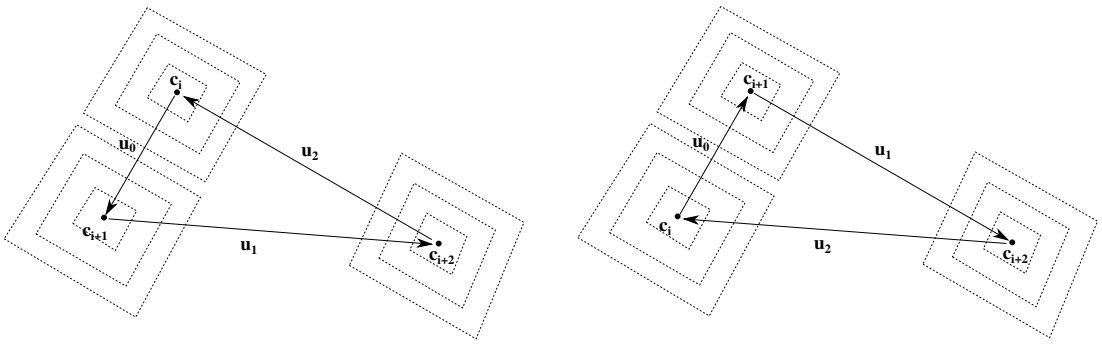

Figura 1.5: Posibles configuraciones de centros posterior a la elección del menor vector $\mathbf{u}_0$.

En caso de que no se encuentre ningún *QlSet* completo se descarta la lista de segmentos y se indica que la determinación de correspondencias falló. Cabe la posibilidad de que los segmentos Debido a posibles oclusionesSi se encuentra un único *QlSet* c

Una vez obtenidos los

Los cuadriláteros de cada QlSet se ordenan según su perímetro.

Se obtiene una lista de tres QlSet la cual se ordena según la disposición espacial de sus centros logrando un ordenamiento que permite definir el sistema de coordenadas previamente explicado.

Los ejes de este sistema de coordenadas nos permite, para cada QlSet, proyectar los vértices sobre sus ejes y según el signo ordenarlos según el orden definido anteriormente.

De esta forma, recorriendo ordenadamente los elementos del marcador, se ordenan los vértices detectados obteniendo una lista de vértices que se corresponde con la lista de vértices del marcador en coordenadas del mundo. Se determinan las correspondencias $\mathbf{M}_i \leftrightarrow \mathbf{m}_i$ necesarias para la estimación de pose.

1.7.3.4. Resultados

Oh sí!

[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.

[10] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, pages 666–673, 1999.

[11] Denis Oberkampf, Daniel F. DeMenthon, and Larry S. Davis. Iterative pose estimation using coplanar feature points. *Comput. Vis. Image Underst.*, 63(3):495–511, may 1996.

[12] Jane Heikkilä and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR 97), June 17-19, 1997, San Juan, Puerto Rico*, page 1106. IEEE Computer Society, 1997.