Universidad de Granada

Grado en Ingeniería Informática. Cuarto.

Cuestionario de teoría: 2.

Nombre de la asignatura:
Visión por Computador. Viernes de 9:30 a 11:30.

Realizado por:
Néstor Rodríguez Vico. DNI: 75573052C.
email: nrv23@correo.ugr.es

ETSIIT
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN.

Granada, 27 de noviembre de 2017.

1. ¿Identificar la/s diferencia/s esencial/es entre el plano afín y el plano proyectivo? ¿Cuáles son sus consecuencias? Justificar la contestación.

En el plano afín existen rectas paralelas, mientras que en el plano proyectivo no. Por lo tanto, si realizamos una tranformación en el plano afín, el paralelismo no se pierde, mientras que en el plano proyectivo sí.

En el libro de *Szeliski* podemos ver como se representa el paralelismo en ambos planos:

affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{3\times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{4\times 4}$	15	straight lines	

2. En coordenadas homogéneas los puntos y rectas del plano se representan por vectores de tres coordenadas (notados x y l respectivamente), de manera que si una recta contiene a un punto se verifica la ecuación $x^T l = 0$. Verificar que en coordenadas homogéneas el vector de la recta definida por dos puntos afines puede calcularse como el producto vectorial de los vectores de ambos puntos (l = x * x'). De igual modo el punto intersección de dos rectas l y l' está dado por x = l * l' ¿Qué conclusiones extrae de las anteriores propiedades de cara a construir un algoritmo que calcule la intersección de dos rectas del plano? Justificar la contestación.

Sabemos que una línea en 2D, l, la podemos representar de la forma $ax + by + c = 0$. De forma análoga, podemos usar coordenada homogéneas para representar a un punto $x = (x_1, y_2)$ de la forma $\vec{p} = (x_1, y_1, 1)$. La condición de que el punto x se encuentra en la recta l es $ax_1 + by_1 + c = 0$ y se puede escribir de la forma $\vec{l} \cdot \vec{p} = 0$. Por lo tanto, tenemos la siguiente equivalencia:

$$\textit{El punto } x \textit{ esta en la linea } l \textit{ en } 2D \Leftrightarrow \vec{x} \textit{ es perpendicular a } \vec{l} \textit{ en } 3D.$$

Como consecuencia tenemos que si l pasa por dos puntos, x, x', tendremos:

$$\vec{l} \cdot \vec{x} \quad y \quad \vec{l} \cdot \vec{x'} \quad \Rightarrow \quad \vec{l} = \vec{x} \times \vec{x'}$$

De esta misma idea tenemos que si un punto x es la intersección entre dos líneas l_1, l_2, tenemos que $\vec{x} = \vec{l_1} \times \vec{l_2}$.

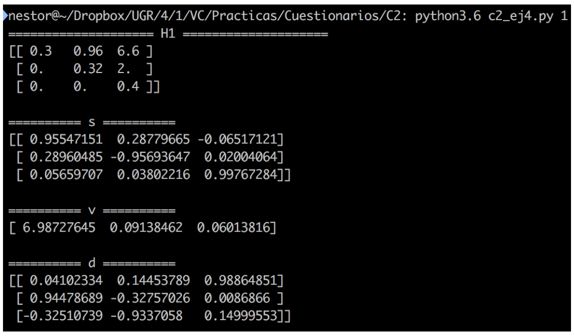
3. Defina una homografía que haga que el punto (2,0,3) del plano proyectivo finito en el plano transformado esté en el infinito. Si recta del infinito l se define como aquella que contiene a todos los puntos del infinito y por tanto debe verificarse que $x^T l = 0$ para todos los puntos del infinito ¿Cuál son las

coordenadas del vector l? Justificar la respuesta.

4. Descomponer en composición de movimientos elementales (traslación, giro, escala, cizalla, proyectivo) cada una de las matrices de las siguientes homografías H1, H2y H3. Justificar las descomposiciones.

Si pensamos un poco en que propiedades conocemos de las matrices, la primera que se nos ocurre es que podemos aplicar *SVD* a cada una de las homografías *H1*, *H2* y *H3*. Por teoría sabemos que, la técnica *SVD* descompone una matriz dada, M, en tres matrices de la forma $U\Sigma V^T$, donde U y V son matrices ortogonales y Σ es una matriz diagonal sin valores negativos. Las matrices ortogonales U y V son matrices que aplican una rotación y la matriz Σ, al ser una matriz diagonal, se trata de una matriz que aplica un escalado. Sabiendo esto, podemos descomponer cualquier homografía en tres matrices aplicando *SVD* y convertir cualquier conjunto de transformaciones en una rotación, seguida de un escalado y finalmente una otra rotación.

Para realizar este proceso de una forma más automatizada, he escrito un pequeño programa en Python que realiza este proceso. Lo primero que hago es calcular *H1*, *H2* y *H3* y luego aplicar *SVD* a las tres y visualizar los resultados. El resultado es el siguiente:

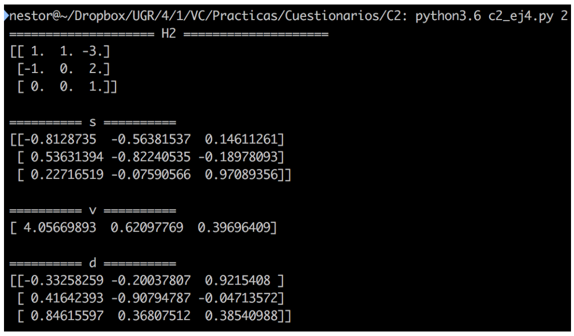


```
nestor@~/Dropbox/UGR/4/1/VC/Practicas/Cuestionarios/C2: python3.6 c2_ej4.py 1
==================== H1 ====================
[[ 0.3   0.96  6.6 ]
 [ 0.    0.32  2.  ]
 [ 0.    0.    0.4 ]]

========== s ==========
[[ 0.95547151  0.28779665 -0.06517121]
 [ 0.28960485 -0.95693647  0.02004064]
 [ 0.05659707  0.03802216  0.99767284]]

========== v ==========
[ 6.98727645  0.09138462  0.06013816]

========== d ==========
[[ 0.04102334  0.14453789  0.98864851]
 [ 0.94478689 -0.32757026  0.0086866 ]
 [-0.32510739 -0.9337058   0.14999553]]
```

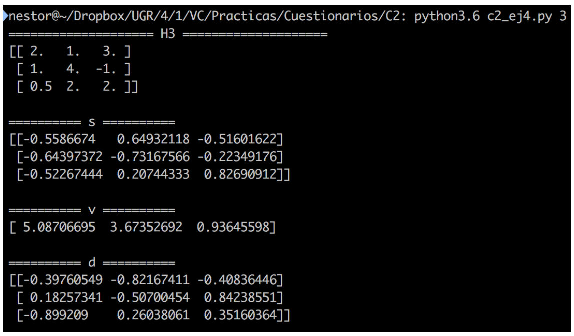


```
nestor@~/Dropbox/UGR/4/1/VC/Practicas/Cuestionarios/C2: python3.6 c2_ej4.py 2
==================== H2 ====================
[[ 1.  1. -3.]
 [-1.  0.  2.]
 [ 0.  0.  1.]]

========== s ==========
[[-0.8128735  -0.56381537  0.14611261]
 [ 0.53631394 -0.82240535 -0.18978093]
 [ 0.22716519 -0.07590566  0.97089356]]

========== v ==========
[ 4.05669893  0.62097769  0.39696409]

========== d ==========
[[-0.33258259 -0.20037807  0.9215408 ]
 [ 0.41642393 -0.90794787 -0.04713572]
 [ 0.84615597  0.36807512  0.38540988]]
```



```
nestor@~/Dropbox/UGR/4/1/VC/Practicas/Cuestionarios/C2: python3.6 c2_ej4.py 3
==================== H3 ====================
[[ 2.   1.   3. ]
 [ 1.   4.  -1. ]
 [ 0.5  2.   2. ]]

========== s ==========
[[-0.5586674   0.64932118 -0.51601622]
 [-0.64397372 -0.73167566 -0.22349176]
 [-0.52267444  0.20744333  0.82690912]]

========== v ==========
[ 5.08706695  3.67352692  0.93645598]

========== d ==========
[[-0.39760549 -0.82167411 -0.40836446]
 [ 0.18257341 -0.50700454  0.84238551]
 [-0.899209    0.26038061  0.35160364]]
```

Podemos ver este proceso en el este enlace[1].

5. ¿Cuáles son las propiedades necesarias y suficientes para que una matriz defina una homografía entre planos? Justificar la respuesta.

[1] Enlace clickeable.

Una homografía es un isomorfismo entre espacios proyectivos. La única propiedad que debe cumplir es que sea una matriz de tamaño 3x3 y que sea invertible, es decir, con determinante distinto de cero. Como bien sabemos y hemos hecho en prácticas, una homografía lleva puntos de un plano p_1 a otro p_2, pero en esa dirección. Si queremos obtener la homografía que lleva puntos del plano p_1 al plano p_2 devemos realizar la inversa de la homografía que ya teníamos. Es decir:

$$H_{p_1 \to p_2} = H_{p_2 \to p_1}^{-1}$$

6. ¿Qué propiedades de la geometría de un plano quedan invariantes si se aplica una homografía general sobre él? Justificar la respuesta.

Esta explicación ha sido sacada del libro *Multiple View Geometry in Computer Vision* de *Richard Hartley*. Debemos saber que una homografía general es una matrix con 8 grados de libertad. Por lo tanto, las propiedades que quedan invariantes son:

- Colinealidad: La colinealidad consiste en que, si varios puntos están alineados en la imagen original, lo seguirán estando tras aplicar la homografía.
- Concurrencia: Sabiendo que se cumple la colinealidad, es fácil saber que se cumple la concurrencia. Esto se debe a que si dos rectas tienen un punto común, el punto que obtenemos tras aplicar la homografía estará en las rectas obtenidas tras aplicar la homografía.
- Orden de contacto (intersección, tangencia e inflexión).
- Discontinuidades tangenciales.
- Ratio cruzado: Tres puntos distintos a, b y c en una línea proyectiva sobre un campo F forman un marco proyectivo de esta línea. Por lo tanto, hay una homografía única h de esta línea en $F \cup \infty$ que lleva a a infinito, b a 0 y c a 1. Dado un cuarto punto de la misma recta, el ratio cruzado de los puntos a, b, c y d, denotado como $[a, b; c, d]$ es el elemento $h(d)$ de $F \cup \infty$. En otras palabras, si d tiene coordenadas homogéneas $[k : 1]$ sobre el marco proyectivo (a, b, c), entonces $[a, b; c, d] = k$.

7. ¿Cuál es la deformación geométrica más fuerte que se puede producir sobre la imagen de un plano por el punto de vista de la cámara? Justificar la respuesta.

Como hemos visto en clase, la deformación geométrica más fuerte es la proyección. Podemos garantizar que si tenemos una recta, esta lo seguirá siendo, pero lo que si perdemos es el paralelismo, ya que los puntos que estaban en el infinito, dejan de estarlo y, por lo tanto, da la sensación de que dos rectas paralelas se cortarán en algún momento.

La forma más clara de ver este efecto es en una foto de las vías del tren como la que podemos ver a continuación[2]:

Podemos ver que, aunque sabemos que las vías del tren son paralelas, la imagen nos da la sensación de que, en un punto, las vías se cruzan.

8. ¿Qué información de la imagen usa el detector de Harris para seleccionar puntos? ¿El detector de Harris detecta patrones geométricos o fotométricos? Justificar la contestación.

El detector de Harris es un detector que busca esquinas. Este detector lo que hace es mirar los gradientes de las derivadas de la imagen. La idea es calcular el valor Harris de cada píxel de la imagen. Para ello calculamos primero la matriz M como:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

donde *w(x,y)* representa la sumatoria de la ventana en la posición *(x, y)*, I_x representa la derivada de la imagen en función de x e I_y representa la derivada de la imagen en función de y.

[2]Imagen obtenida de Wikipedia.

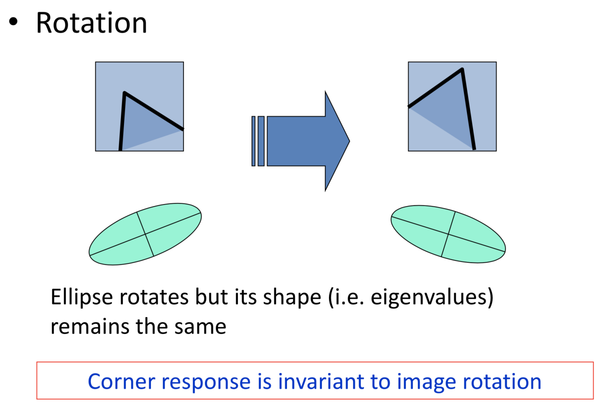
Una vez tenemos la matriz M, aplicamos el criterio de Harris:

$$R = det(M) - k(Trace(M))^2$$

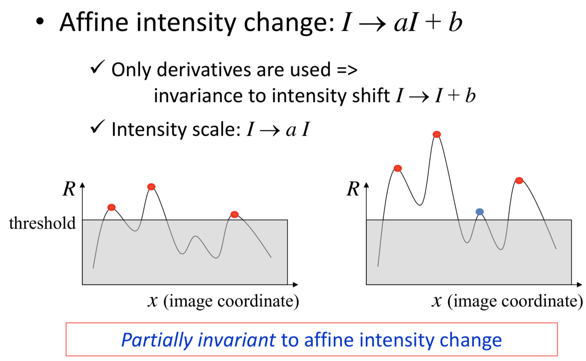
Finalmente, los píxeles con un alto valor de respuesta (valor de la matriz R) serán los puntos Harris a tener en cuenta (tras hacer lo que hemos hecho en prácticas, la supresión de valores no máximos).

9. ¿Sería adecuado usar como descriptor de un punto Harris los valores de los píxeles de su región de soporte? En caso positivo identificar cuando y justificar la respuesta.

Debemos partir de la idea de que un punto Harris se ha convertido en punto Harris por el valor de su región de soporte (valor de la matriz R del ejercicio anterior). La idea de usar un descriptor es su uso posterior para poder establecer correspondencias entre dos imágenes, así que, ¿es realmente bueno este descriptor? Como podemos ver en las transparencias de teoría este detector es invariante a las rotaciones como podemos ver en la siguiente imagen[3]:

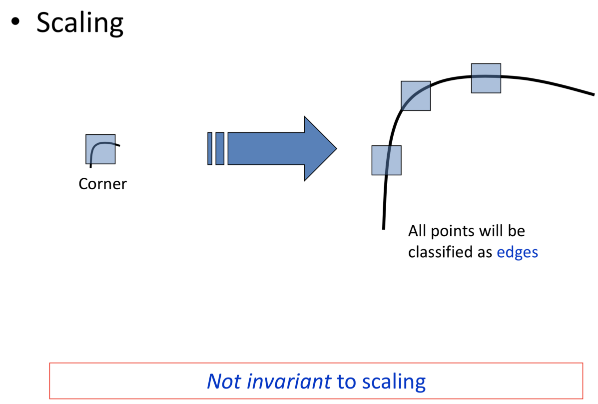


También sabemos que es parcialmente invariante a los cambios de intensidad, tal y como podemos ver en la siguiente imagen:



[3]Todas las imágenes son sacadas de las transparencias de teoría.

Pero, sin embargo, como podemos ver en la siguiente imagen, no es invariante frente a los escalados:



Por lo tanto, no es buena idea usar dicho criterio, ya que si tenemos que establecer correspondencias entre dos imagenes y una de ellas incluye un escalado, no obtendremos unos resultados del todo buenos.

10. ¿Qué información de la imagen se codifica en el descriptor de SIFT? Justificar la contestación.

Una vez más nos encontramos con que se usan los gradientes de la imagen. El proceso lo podemos ver en el libro *Computer Vision: Algorithms and Applications* de *Richard Szeliski*.

Lo que hacemos es calcular el gradiente para cada píxel dentro de una ventana de tamaño 16x16 centrada en el *KeyPoint* detectado. A continuación se reducen las magnitudes de los gradientes usando una función Gausiana (representada por el círculo amarillo de la foto 1 para reducir la influencia de los gradientes centrales.

Luego obtenemos un histograma de orientación de gradiente para cada cuadrante de tamaño 4x4 añadiendo el valor de gradiente ponderado a uno de los ocho intervalos de histograma de orientación. Para reducir los efectos de la ubicación y la desestimación de la orientación dominante, cada una de las 256 magnitudes de gradiente ponderadas originales se agrega suavemente a los intervalos de histograma de 2x2x2 mediante interpolación trilineal.

Los 128 valores no negativos resultantes forman una versión bruta del vector de descriptor SIFT. Para reducir los efectos del contraste o la ganancia, se normaliza para que la longitud sea la unidad. Para hacer más robusto el descriptor a otras variaciones fotométricas, los valores se recortan a 0.2 y el vector resultante se normaliza de nuevo.

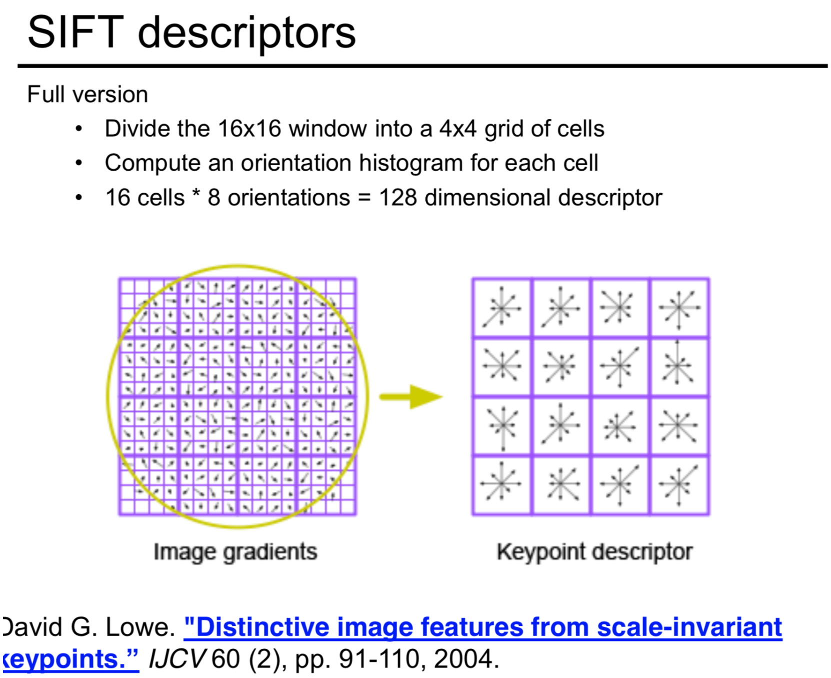


Figura 1: Descriptor SIFT.

11. Describa un par de criterios que sirvan para establecer correspondencias (matching) entre descriptores de regiones extraídos de dos imágenes. Justificar la idoneidad de los mismos.

En clase hemos visto dos maneras de hacerlo:

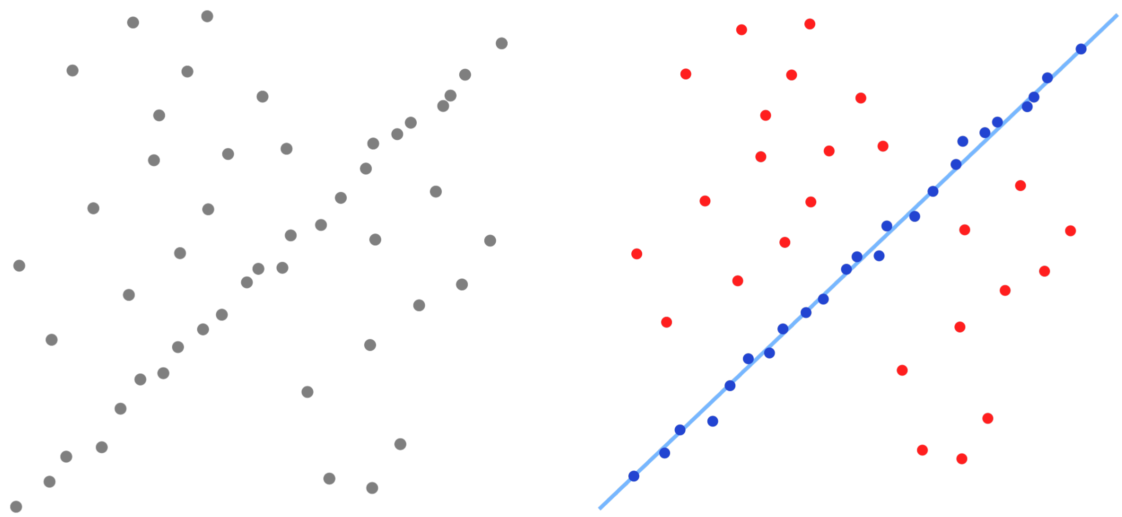
- Fuerza bruta: La idea es encontrar para cada punto de la imagen fuente un punto correspondiente en la imagen destino. Para ello calcularemos la distancia entre el punto fuente y todos los puntos objetivos y nos quedaremos con el que nos proporcione una menor distancia. La distancia se puede calcular usando la distancia *euclidea* o la *norma 1*. Esta idea tiene una ventaja, es simple y sencilla. Pero tiene un problema, puede fallar si las imágenes son patrones, tal y como pasaba a las imágenes de la valla del jardín.

- La otra idea es una versión mejorada de la anterior. En vez de calcular un único punto destino para cada punto fuente, calculamos dos (los dos de distancia menor) y diremos que la correspondencia es fiable si la distancia del punto fuente al primer punto destino es menor que un umbral multiplicado por la distancia del punto fuente al segundo punto destino. Es decir, la correspondencia será buena si el punto fuente se parece (tiene una distancia pequeña) bastante al primer punto destino y no se parece (tiene una distancia grande) al segundo punto destino. De esta forma podemos evitar el problema de las correspondencias asociadas a los patrones.qu

12. Cual es el objetivo principal en el uso de la técnica RANSAC. Justificar la respuesta.

Este algoritmo se usa en el proceso de búsqueda de una homografía que relacione dos imágenes. El problema reside en cuando tenemos “valores atípicos” (*outliers*), que son datos que no encajan en el modelo. Los valores atípicos pueden provenir, por ejemplo, de valores extremos del ruido o de mediciones erróneas o hipótesis incorrectas sobre la interpretación de los datos.

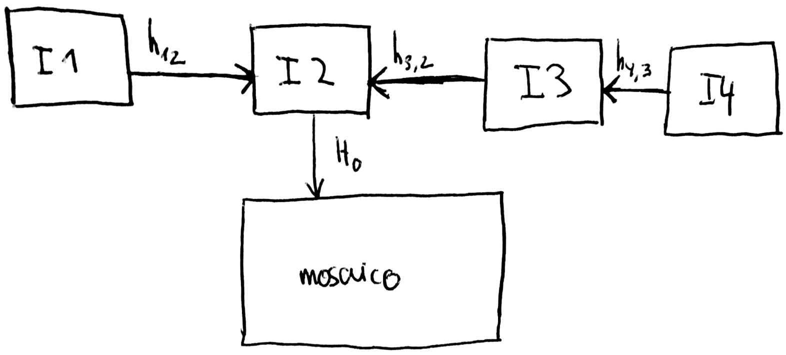
Cómo hemos visto en teoría, usaremos la técnica de mínimos cuadrados para encontrar la homografía correspondiente. Esta técnica intentará ajustar una recta a para todos los puntos proporcionados, inclusos los ya mencionado “valores atípicos”. Aquí es donde aparece *RANSAC*, el cual intenta ajustar una recta pero teniendo sólo en cuenta los puntos válidos (*inliers*). En la siguiente imágenes[4] podemos ver como tenemos mucho “valores atípicos” (primera imagen) pero que *RANSAC* ajusta una recta bastante buena (segunda imagen).



13. ¿Si tengo 4 imágenes de una escena de manera que se solapan la 1-2, 2-3 y 3-4. ¿Cuál es el número mínimo de puntos en correspondencias necesarios para montar un mosaico? Justificar la respuesta.

Nosotros queremos montar lo siguiente:

[4]Imagenes sacadas de la Wikipedia.



En clase vimos que necesitamos 4 correspondencias para cada par de imágenes para obtener una homografía. Nosotros tenemos que calcular 3 homografías (la que va de la imagen 1 a la 2, la que va de la imagen 3 a la imagen 2 y la que va de la imagen 4 a la 3). Por lo tanto, necesitamos 12 correspondencias. Por definición, una correspondencia es una relación entre 2 puntos, así que necesitamos 24 puntos.

Si queremos llevar todas las imágenes a un mosaico resultado necesitaremos la homografía H_0. Como hemos visto en teoría, nosotros suponemos que la imagen central (la 2 en mi dibujo) no tiene ninguna deformación. Por lo tanto, la homografía H_0 sólo es una traslación. Para realizar una traslación sólo necesitamos una correspondencia, y por lo tanto dos puntos. Así que en total serían 26 puntos.

14. En la confección de un mosaico con proyección rectangular es esperable que aparezcan deformaciones de la realidad. ¿Cuáles y por qué? ¿Bajo qué condiciones esas deformaciones podrían desaparecer? Justificar la respuesta.

Las deformaciones que aparece se dan cuando las imágenes están más alejadas con respecto de la imagen central. Esto se debe a que vamos acumulando el error según nos vamos alejando de la imagen central. Por ello, cuando montamos un mosaico elegimos una imagen del centro como referencia y no una de los extremos, para tratar de minimizar lo máximo el error.

Este tipo deformaciones se puede evitar si las imágenes son tomadas moviendo la cámara sólo en el eje x y el eje y, es decir, a derecha e izquierda y arriba y abajo. Si hacemos esto, las imágenes se pueden proyectar sobre una proyección rectángular sin sufrir tanta deformación. El problema aparece cuando la cámara se rota en el eje z.

Bibliografía [5]:

- Plano proyectivo - Wikipedia.
- Plano afín - Wikipedia.

[5] Algunos elementos de la bibliografía son enlaces que llevan a la página en concreto.

- SVD - Wikipedia
- SVD - StackExchange
- Línea definida por dos puntos - StackExchange
- Homografía - Wikipedia
- Ratio cruzado - Wikipedia
- RANSAC - Wikipedia.
- Computer Vision: Algorithms and Applications - Richard Szeliski
- Multiple View Geometry in Computer Vision - Richard Hartley.
- Transparencias de teoría