

Prácticas de Visión por Computador

Grupo 2

Más Ideas sobre la P2

Pablo Mesejo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial



UNIVERSIDAD
DE GRANADA



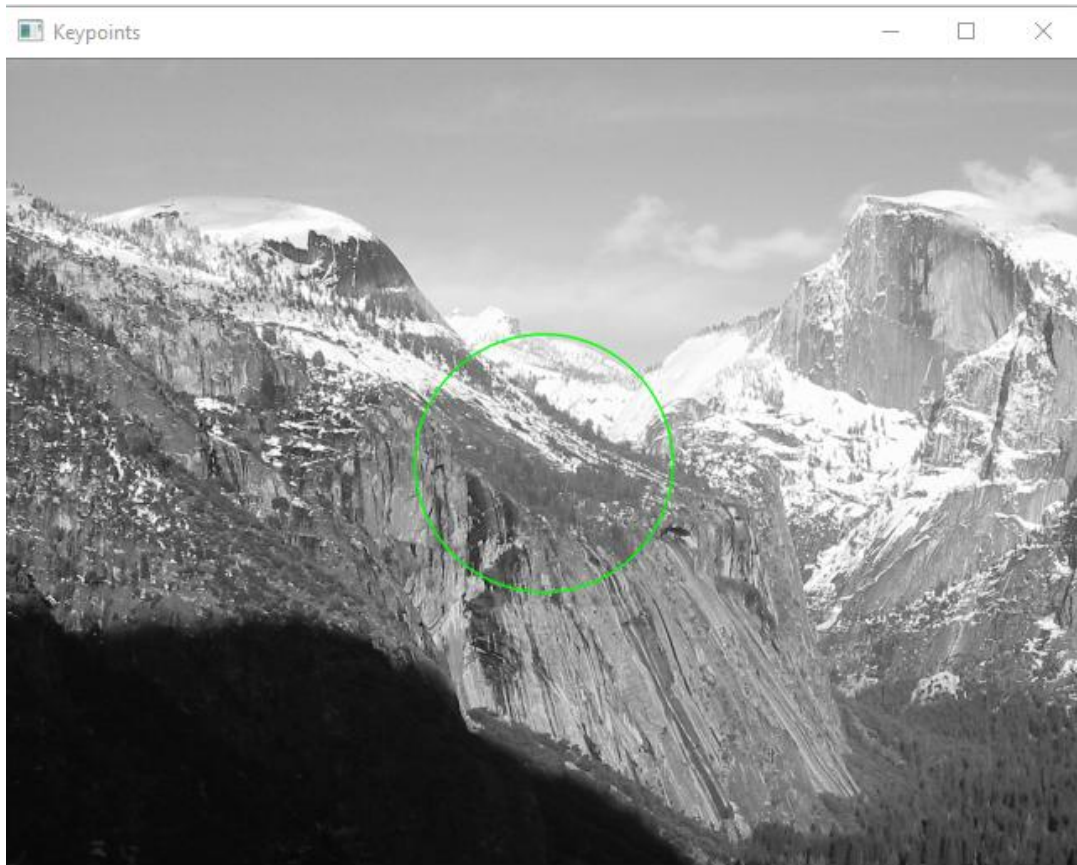
Ejercicio 1.e

«Mostrar la imagen con los extremos locales extraídos usando para ello un círculo de radio 6σ sobre la escala de detección σ »

- ¿No será demasiado grande el keypoint?

Ejercicio 1.e

- El σ_k más grande que podemos referenciar en nuestro caso es 12.8. Si lo multiplicamos por 6, nos da un radio de 76.8 píxeles.
- `cv2.KeyPoint` trabaja con el diámetro, no el radio
→ $76.8 \cdot 2$

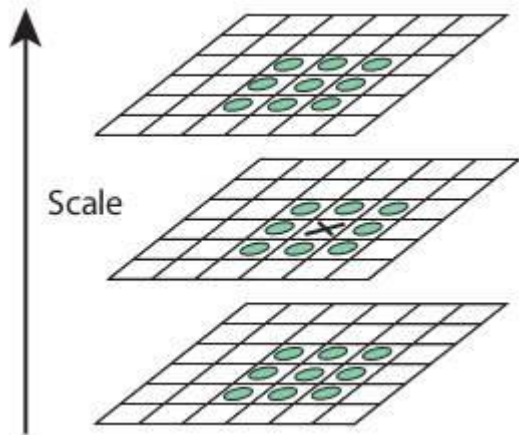


Ejercicio 1.d

- ¿Cómo optimizar la identificación de extremos en la escala Laplaciana?
 - Ignorar aquellos puntos que formen parte del vecindario de un extremo
 - Para ello, necesitaréis una pirámide de máscaras que determinen dónde es necesario buscar

Ejercicio 1.d

- ¿Cómo optimizar la identificación de extremos en la escala Laplaciana?



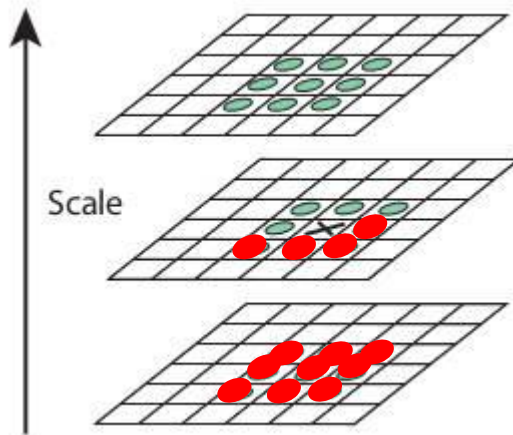
Si **X** es extremo local, no es necesario revisar ninguno de sus **vecinos**

Notas:

- Lo más costoso será la primera octava, al ser la más grande.
- Si sigue siendo muy costoso, una alternativa (a discutir con el profesor) sería emplear entornos más grandes que 3x3x3

Ejercicio 1.d

- ¿Cómo optimizar la identificación de extremos en la escala Laplaciana?



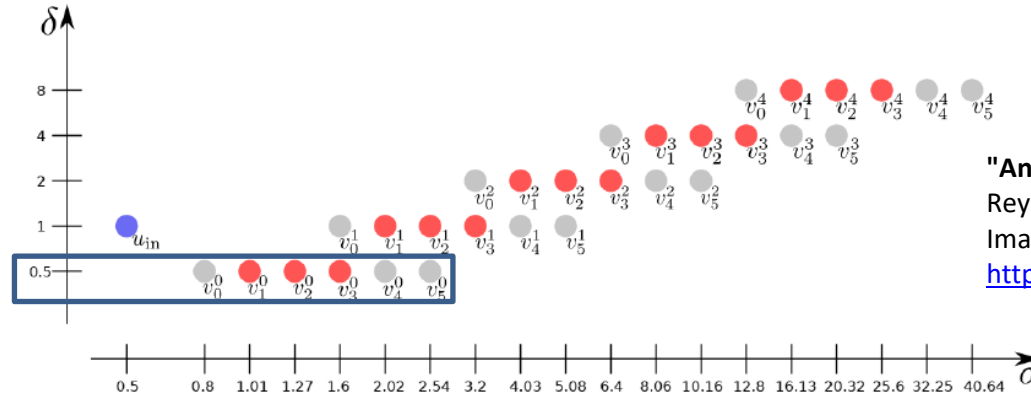
Si **X** es extremo local, no es necesario revisitar ninguno de sus **vecinos**

Si vamos de arriba abajo y de izquierda a derecha, ya no necesitaremos visitar los puntos en **rojo**

Notas:

- Lo más costoso será la primera octava, al ser la más grande.
- Si sigue siendo muy costoso, una alternativa (a discutir con el profesor) sería emplear entornos más grandes que 3x3x3

Interés de detectar keypoints en la octava -1



"Anatomy of the SIFT Method" (Ives
Rey-Otero, Mauricio Delbracio)
Image Processing On Line, 2014
<http://dx.doi.org/10.5201/ipol.2014.82>

- No totalmente claro por qué, en ocasiones, se usa esa octava -1 (o 0)
 - Algunos autores la usan, otros no.
- Principal intuición:
 - Si la imagen es grande, nos da igual esa octava.
 - Si la imagen es pequeña, al suavizar, puede que eliminemos objetos de interés
 - En cambio, si ampliamos e interpolamos, quizás sí podamos detectar esos pequeños objetos de interés.

Ejercicio 2

- La homografía de base (mera traslación) se puede calcular manualmente. No necesitáis calcular keypoints

```
H0 = np.array([[1, 0, ancho_canvas//2 - imagen.shape[1]//2],  
               [0, 1, alto_canvas//2 - imagen.shape[0]//2],  
               [0, 0, 1]])
```

Nota: Probablemente, si el panorama os sale cortado, sea que esta homografía no es correcta o que el canvas no es suficientemente grande...

Ejercicio 2

- Sobre `cv2.warpPerspective()`
 - Aplica una transformación de perspectiva a una imagen

La imagen de salida es nuestro canvas

```
cv2.warpPerspective(image_left, H_left2canvas, (width_canvas, height_canvas), dst=canvas, borderMode =  
cv2.BORDER_TRANSPARENT)
```

$H_left2canvas$ es la homografía resultante de la composición de llevar la imagen de la izquierda al centro ($H_left2center$) y del centro al canvas ($H0$) $\rightarrow H_left2canvas = H0 * H_left2center$

Nota: recordad que el orden que vamos «rellenando» el canvas es importante. Queremos insertar de última la imagen menos transformada.

Ejercicio 2

- Si queréis darle un acabado más bonito al panorama, podéis eliminar las bandas negras sobrantes.
 - Una posibilidad es usar `cv2.boundingRect(img)`

https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html



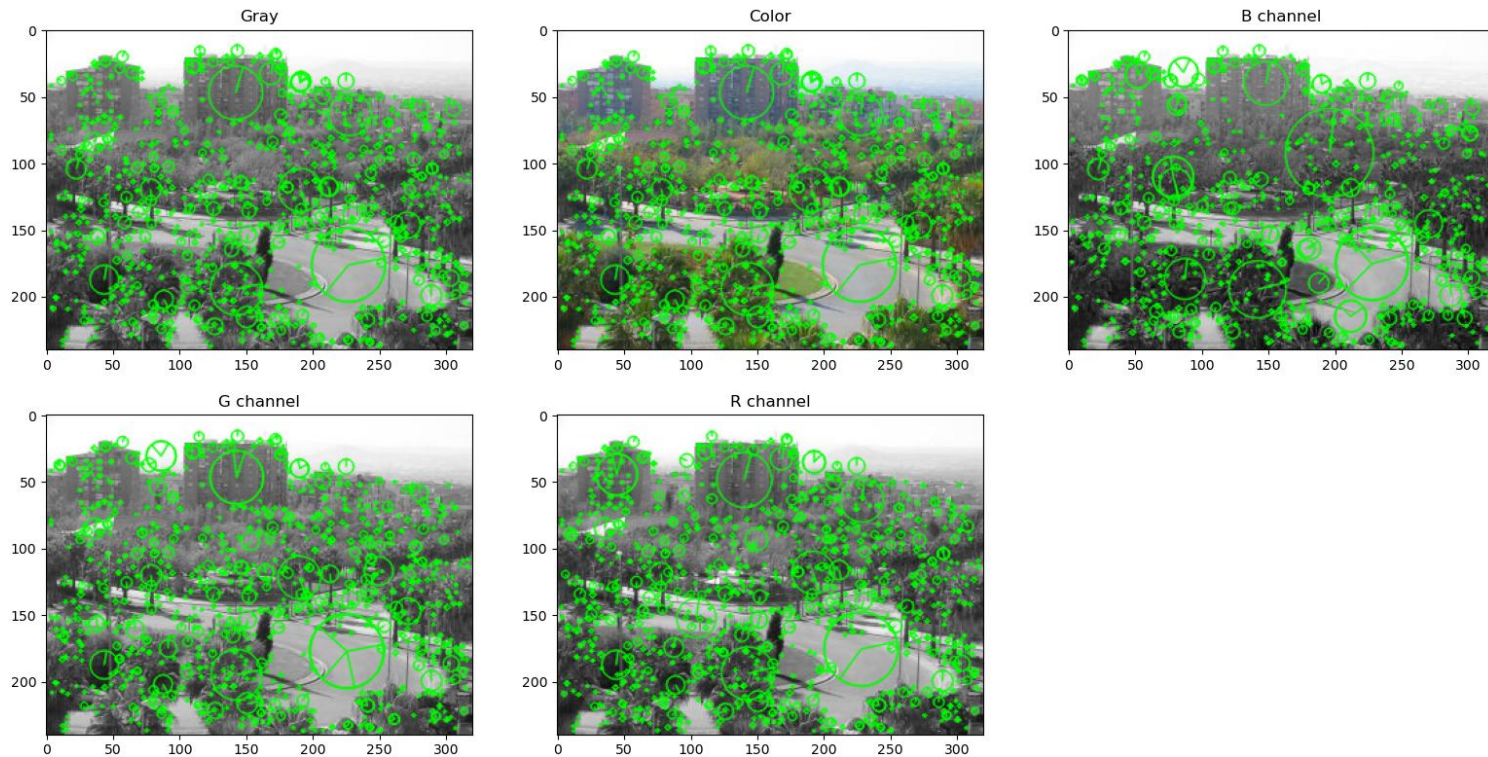
Sobre SIFT y el uso de RGB

- SIFT opera en imágenes en escala de grises
 - La implementación de OpenCV convierte imágenes en color a escala de grises
- Lowe, en las conclusiones de su paper [1], indica lo siguiente:

The features described in this paper use only a monochrome intensity image, so further distinctiveness could be derived from including illumination-invariant color descriptors (Funt and Finlayson, 1995; Brown and Lowe, 2002).

Sobre SIFT y el uso de RGB

- Si se emplea SIFT en las distintas bandas (R, G, B) de una imagen, las diferencias no son especialmente marcadas

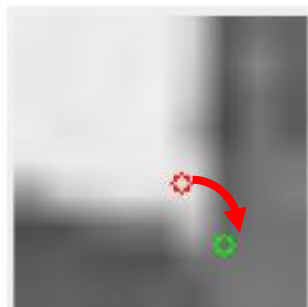


Bonus B1

Interpolar keypoint dentro del “cubo” en que ha sido detectado para moverlo a la mejor localización posible.

- Sección 4 («Accurate keypoint localization»), artículo «Distinctive Image Features from Scale-Invariant Keypoints»
- Sección 3.2 («Extraction of candidate points»), subsección «Keypoint position refinement», artículo «Anatomy of the SIFT Method»

Offset vector to be added to the initial estimate



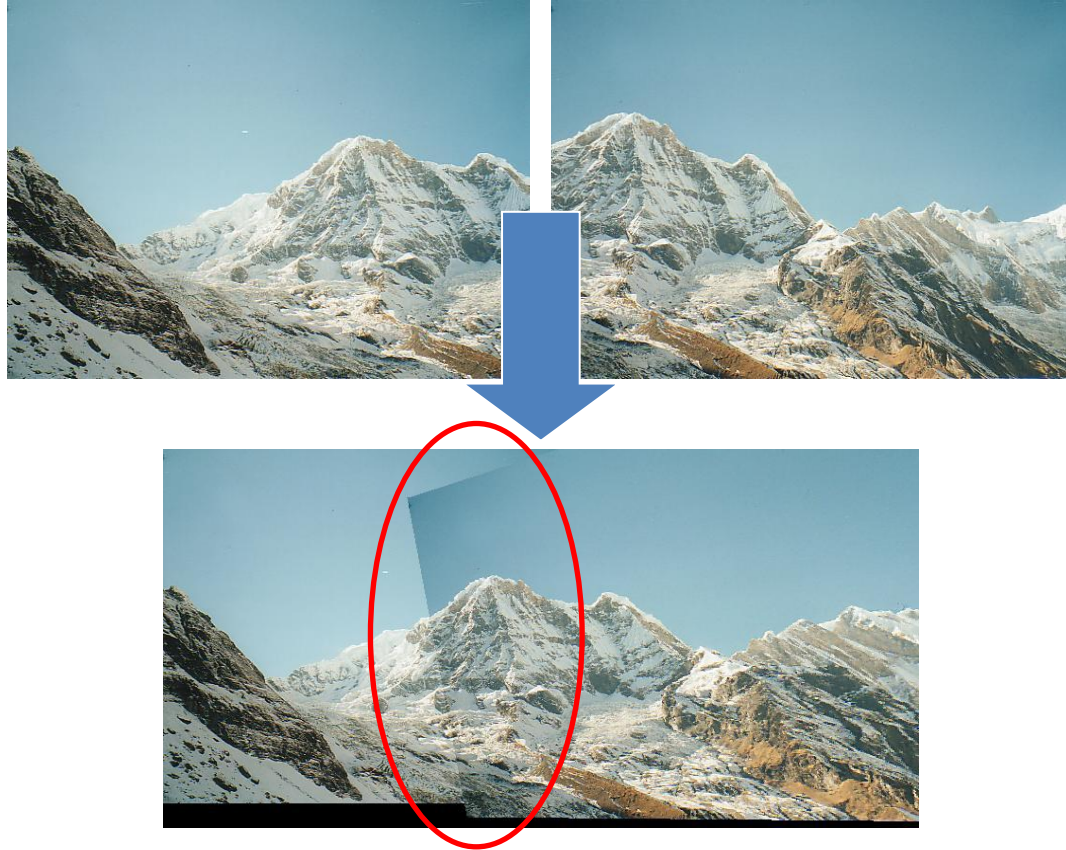
$$\hat{x} = -\frac{\partial^2 H}{\partial \mathbf{x}^2}^{-1} \frac{\partial H}{\partial \mathbf{x}}$$

Jacobian

Inverse Hessian

siendo $H(x, y, \sigma)$ el extremo local encontrado en el espacio de escalas Laplaciano

Nuevo bonus B2.b y B2.c



Nuevo bonus B2.b y B2.c



Nuevo bonus B2.b y B2.c

- Brown, M., & Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1), 59-73.
 - Y, más concretamente, la sección 7 de dicho paper sobre “**Multi-Band Blending**”

1. Insert the blending process just after the transportation of each image into the Mosaic.
2. For every two consecutive aligned images (I1,I2):
 1. Extract the smallest rectangular region that contains each image in the Mosaic (mosaic images)
 2. Blend the images: $I2 = \text{Blend}(I1, I2, \text{shift})$, shift: translation vector between images
 3. Copy the I2-blended image into the Mosaic.

Blending-Algorithm: Input (I1,I2,shift), output: image

1. Build Laplacian pyramids for I1 and I2
2. Compute the binary weighting masks for each image
3. Compute GPs for both weighting-masks
4. For each layer in the LPs:
 1. Build a new layer blending the bands I1 and the shifted-I2 applying the weighting masks
5. Reconstruct the blended image
6. Give as output the I2 region

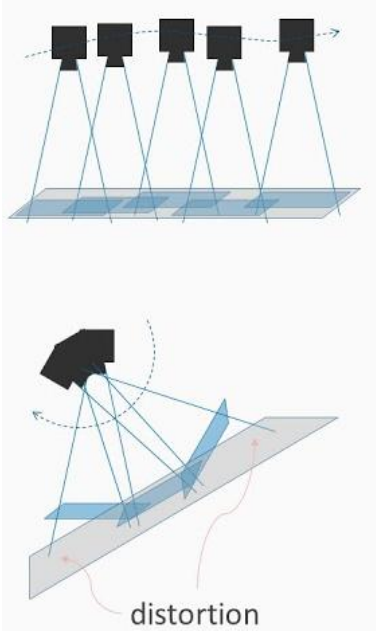
Lectura recomendable!

La respuesta a algunas de las dudas que pueda generar este paper se pueden encontrar en Burt & Adelson (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4), 217–236.

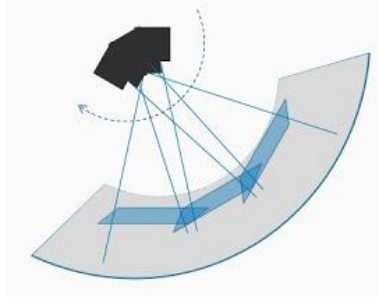
Nuevo bonus B2.c

Szeliski, Richard, and Heung-Yeung Shum. "Creating full view panoramic image mosaics and environment maps." *Proceedings of the 24th annual conference on computer graphics and interactive techniques*. 1997.

Proyección lineal



Proyección cilíndrica



Proyección esférica



Antes montábamos el mosaico sobre una superficie plana, y ahora dicha superficie es esférica

Nuevo bonus B2.c



Figure 4: Tessellated spherical panorama covering the north pole (constructed from 54 images). The white triangles at the top are the parts of the texture map not covered in the 3D tessellated globe model (due to triangular elements at the poles).

Prácticas de Visión por Computador

Grupo 2

Más Ideas sobre la P2

Pablo Mesejo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial



UNIVERSIDAD
DE GRANADA

