

Alineación de imágenes

Definición

- **Alineación de imágenes.** Proceso en el que se ajustan varias imágenes para que coincidan o se alineen entre sí en términos de geometría.
- El objetivo es lograr que los puntos de diferentes imágenes coincidan para que se puedan analizar juntas o seguir procesando.

Propósito

- **Pegado de imágenes:** alinear imágenes superpuestas para crear una imagen panorámica más grande.



+



=



Propósito

- **Reconstrucción 3D:** alinear imágenes tomadas desde diferentes puntos de vista para reconstruir un objeto en 3D.
- **Imágenes médicas:** alinear de imágenes de diferentes fuentes para analizar mejor las características anatómicas o detectar anomalías.
- **Seguimiento de objetos:** alinear frames de video para rastrear el movimiento de objetos a lo largo del tiempo.
- **Realidad aumentada (RA):** alinear elementos virtuales con objetos del mundo real en video.

Propósito

- **Fusión de imágenes multimodales:** alinear de imágenes de diferentes sensores para obtener información mejorada.

Métodos

1. Métodos directos:

- Operan sobre los píxeles sin detección de características.
- Utilizan técnicas de optimización para minimizar las diferencias entre imágenes.

2. Métodos basados en características:

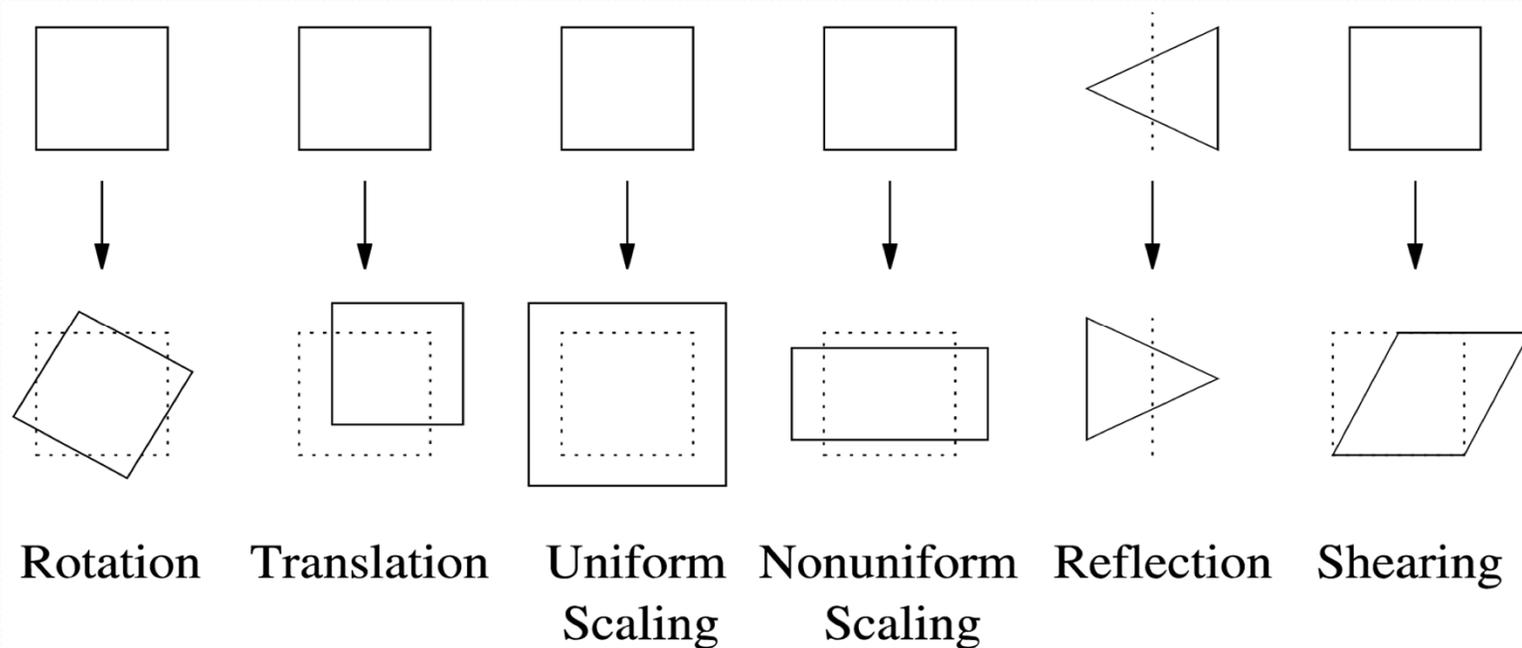
- Detectan keypoints (por ejemplo, esquinas, bordes) y los utilizan para calcular transformaciones.
- Se utilizan técnicas como SIFT y ORB para la detección de características.



Modelos de movimiento

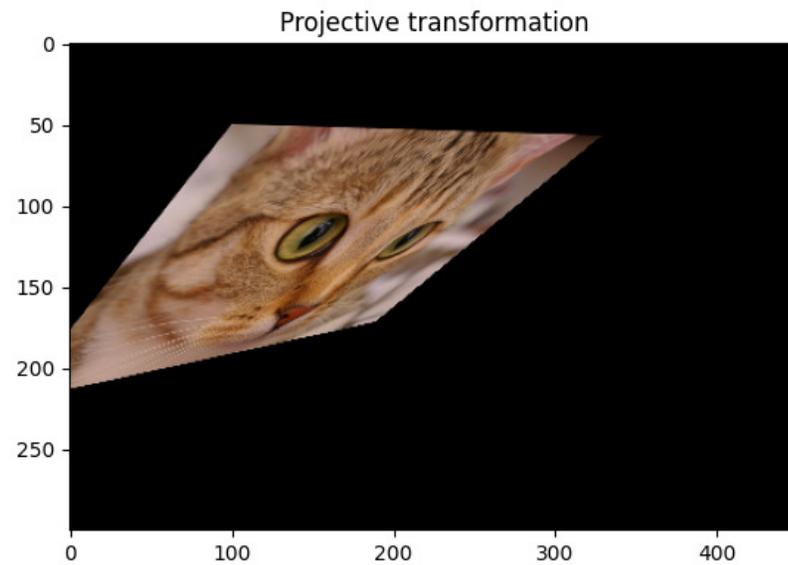
- Transformaciones afines. Preservan líneas rectas y líneas paralelas, pero no distancias ni ángulos.
- Homografía. Solo preserva líneas rectas.

Transformaciones afines



Fuente: [https://www.cs.drexel.edu/~david/Classes/CS536/Lectures/L-18 Transformations.pdf](https://www.cs.drexel.edu/~david/Classes/CS536/Lectures/L-18_Transformations.pdf)

Homografía



Fuente: https://scikit-image.org/docs/stable/auto_examples/transform/plot_transform_types.html

Traslación

- Traslación por un vector (d_x, d_y) :

- $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \end{pmatrix}$$

Rotación

- Rotación por un ángulo α con el origen como centro de rotación:

- $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

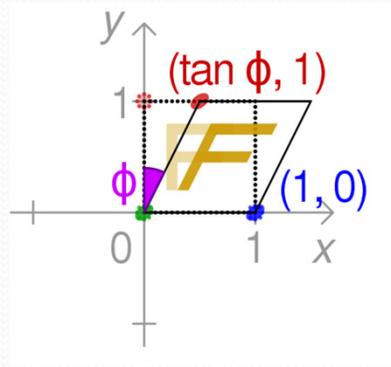
Rotación

- Rotación por un ángulo α con un punto arbitrario (x_c, y_c) como centro de rotación:

- $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Shear

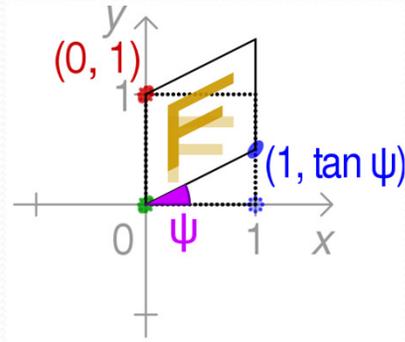
- Shear en la dirección x con un ángulo ϕ :



- $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & \tan \phi & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Shear

- Shear en la dirección y con un ángulo ψ :



- $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \tan \psi & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Homografía

- Dos imágenes de una escena están relacionadas por una homografía bajo dos condiciones.
 1. Las dos imágenes son las de un plano.
 2. Las dos imágenes se adquirieron rotando la cámara sobre su eje óptico.
- Una homografía es una matriz de 3×3 :

- $$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}$$

Homografía

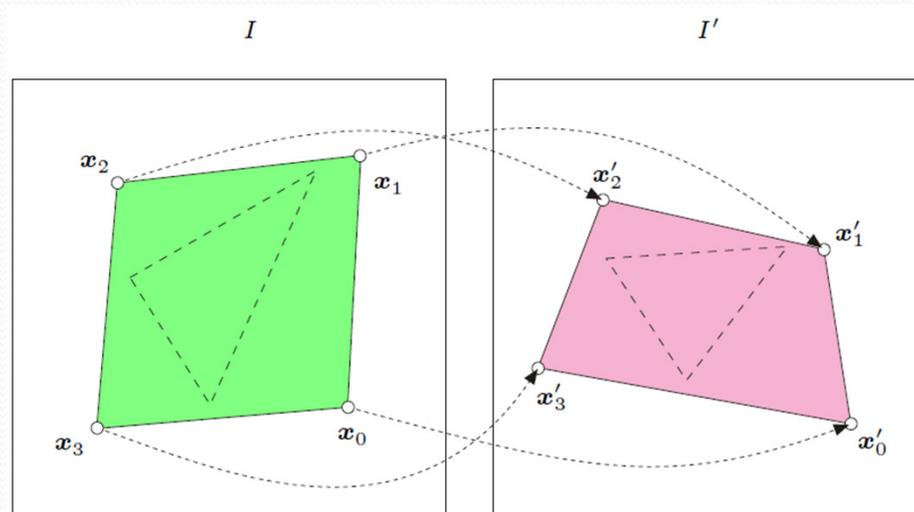
- Sea (x_1, y_1) un punto en la primera imagen y (x_2, y_2) las coordenadas del mismo punto físico en la segunda imagen.
- La homografía H los relaciona de la siguiente manera:

- $$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

- La homografía, se aplica a todos los píxeles de una imagen para obtener una imagen deformada que esté alineada con la segunda imagen.

Homografía

- Para determinarla se necesitan al menos 4 puntos.



Fuente: DIP-Java, p. 609.

Homografía

- Se puede obtener resolviendo un sistema de 8 ecuaciones con 8 incógnitas.

$$\begin{pmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0x'_0 & -y_0y'_0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0y'_0 & -y_0y'_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1y'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2y'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3y'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \end{pmatrix} \cdot \begin{pmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{20} \\ a_{21} \end{pmatrix} = \begin{pmatrix} x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{pmatrix}, \quad (21.38)$$

Fuente: DIP-Java, p. 609.

Homografía

- O se puede usar OpenCV usando el método `cv2.findHomography()`.

Calcula la homografía

```
h, status = cv2.findHomography(pts_src, pts_dst)
```

- La homografía se aplica a una imagen usando el método `cv2.warpPerspective()`.

Deforma la imagen fuente de acuerdo a la homografía

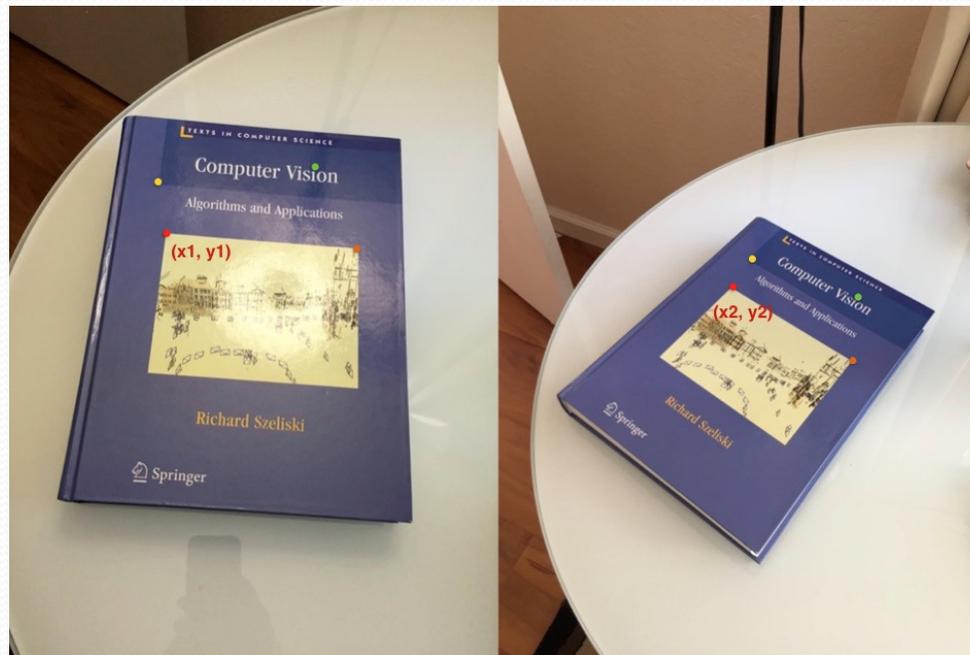
```
im_out = cv2.warpPerspective(im_src, h, (im_dst.shape[1],  
                                         im_dst.shape[0]))
```

Aplicaciones de la homografía

- Rectificación de imágenes.
- Corrección de perspectiva.
- Carteles virtuales.
- Matching de imágenes cuando el patrón está rotado.

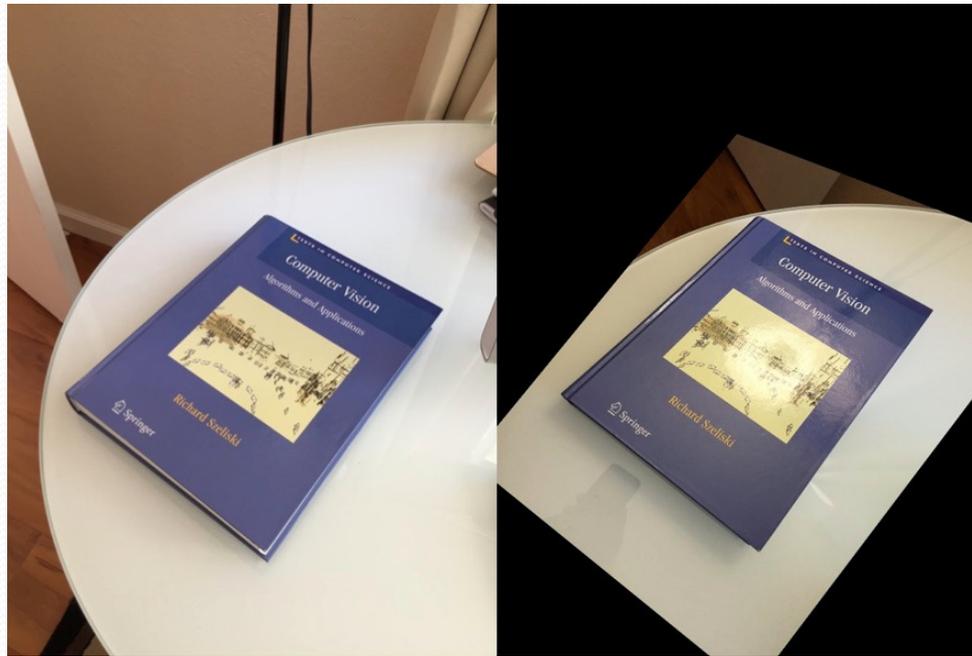
Rectificación de imágenes

- Alinear el primer libro con el segundo.



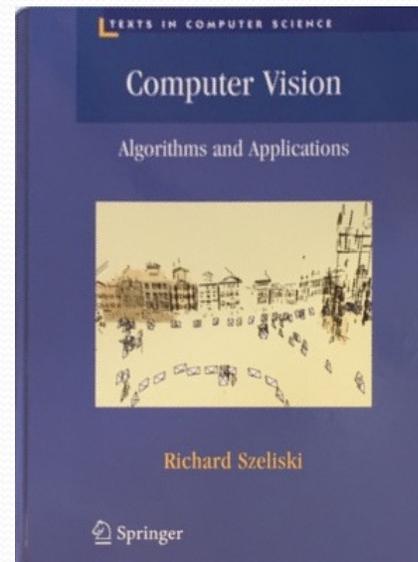
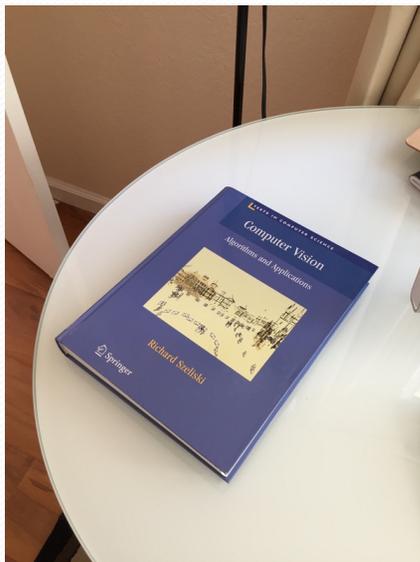
Rectificación de imágenes

- Resultado:



Corrección de perspectiva

- Aplicar una transformación a la imagen de la izquierda para obtener la imagen de la derecha.



Carteles virtuales

- Insertar la imagen de la izquierda en la imagen de la derecha, de tal forma que sustituya el cartel de Motown.



Fuentes: [https://upload.wikimedia.org/wikipedia/en/c/c0/Les Horribles Cernettes in 1992.jpg](https://upload.wikimedia.org/wikipedia/en/c/c0/Les_Horribles_Cernettes_in_1992.jpg) y [https://upload.wikimedia.org/wikipedia/commons/thumb/c/c0/1 times square night 2013.jpg/800px-1 times square night 2013.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/c/c0/1_times_square_night_2013.jpg/800px-1_times_square_night_2013.jpg)

Carteles virtuales

- Resultado:



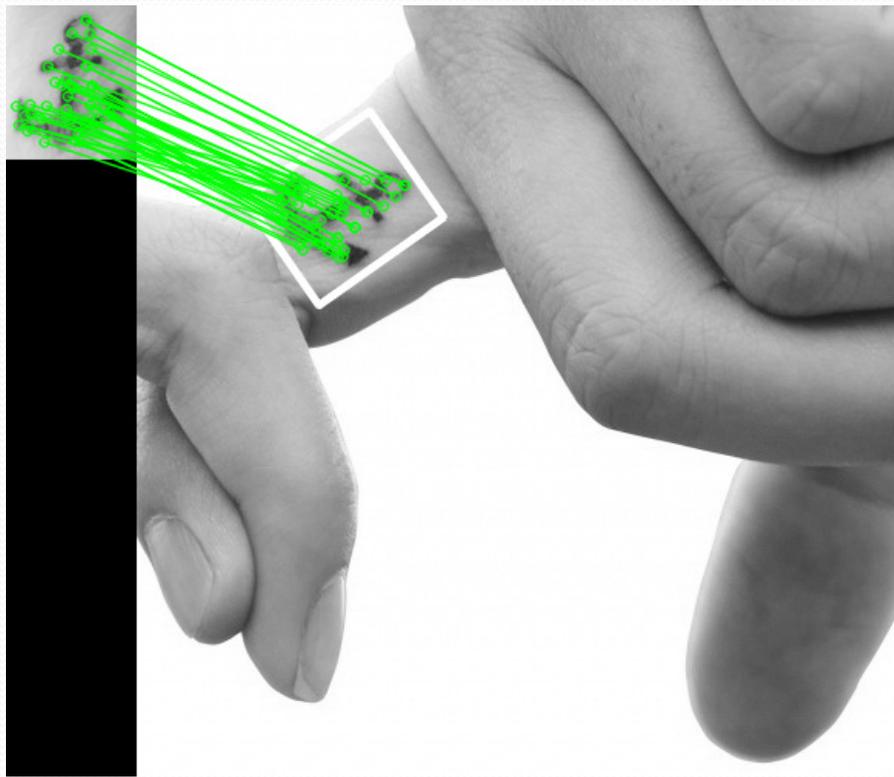
Universidad de Sonora

Matching de imágenes

- Buscar la imagen de la izquierda en la imagen de la derecha.



Resultado



Alineación de imágenes

- Métodos directos.
- Métodos basados en características.

Métodos directos

- **SSD (suma de diferencias al cuadrado)**. Desliza una imagen sobre la otra calculando la suma de diferencias al cuadrado de las intensidades.
- **Correlación cruzada**. Desliza una imagen sobre la otra calculando el coeficiente de correlación.
- **ECC (coeficiente de correlación mejorado)**:
 - Normaliza la imagen ($\mu = 0, \sigma^2 = 1$).
 - Calcula la correlación cruzada.
 - Esto hace que ECC sea robusto a cambios de iluminación.

Ejemplo con ECC

- Sergey Prokudin-Gorskii fue un pionero de las fotografías a color a principios del siglo 20.
- Su idea fue tomar tres fotografías de cada escena usando un filtro rojo, uno verde y uno azul.



Ejemplo con ECC

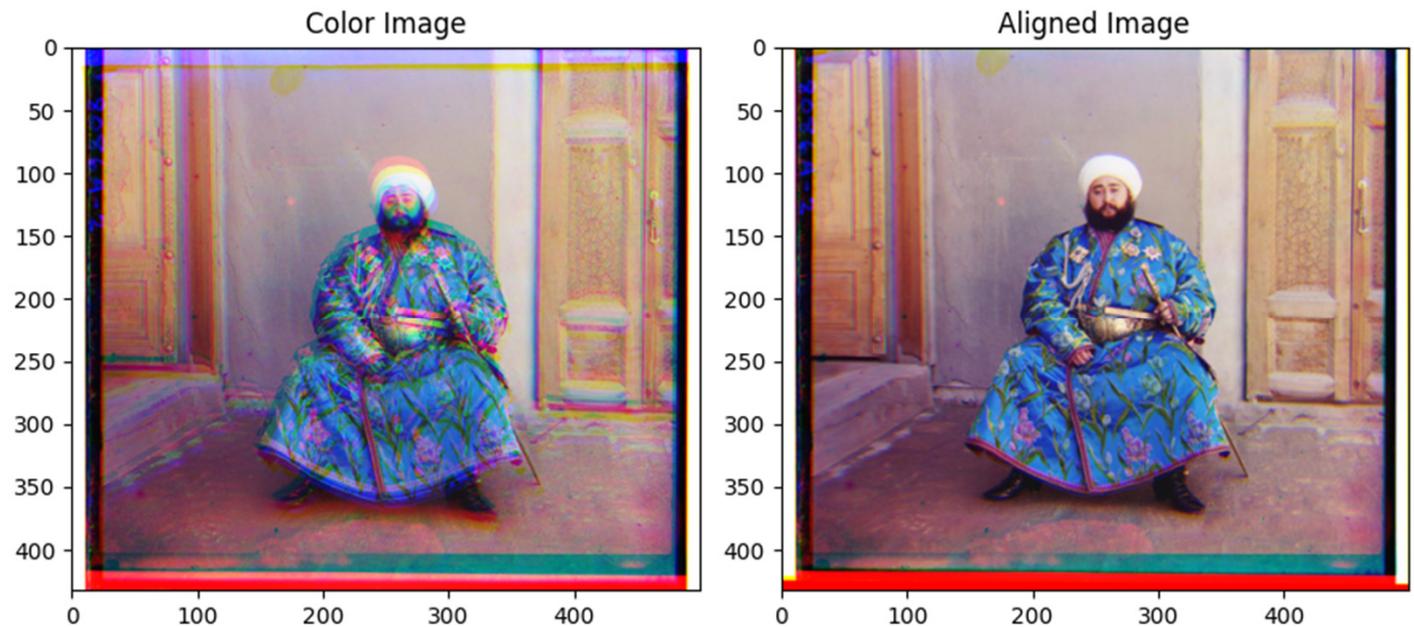
- La colección digitalizada Prokudin-Gorskii está disponible en <https://www.loc.gov/exhibits/empire/gorskii.html>
- El problema es que los frames están desalineados.



Ejemplo con ECC

- Solución:
- Se deja un canal fijo y los otros 2 se alinean con el.
- Para cada canal:
 - Se obtiene la homografía con `cv2.findTransformECC()`.
 - Se aplica la homografía con el método `cv2.warpPerspective()`.

Resultado



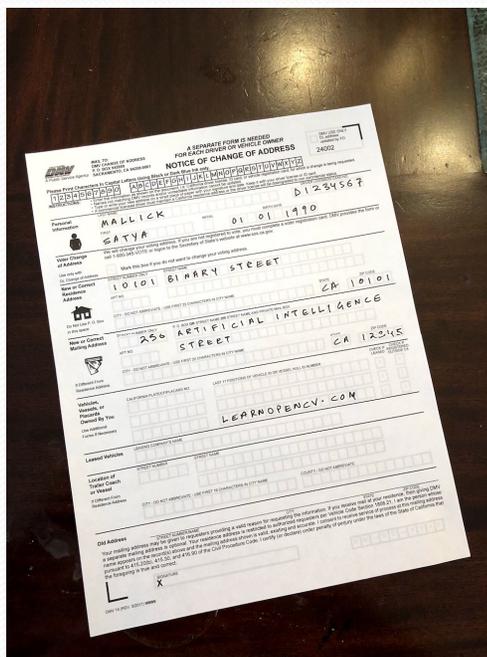
- **Nota:** en una laptop Dell, con Core i5 y 8 GB de memoria, la alineación se tarda entre 3 y 4 minutos.

Métodos basados en características

1. Obtener los keypoints de las dos imágenes con ORB o SIFT.
2. Obtener la lista de coincidencias entre los dos arreglos de keypoints.
3. Ordenar las coincidencias por distancia.
4. [Opcional] Quitar las coincidencias con distancia grande.
5. Extraer de los keypoints las coordenadas.
6. Calcular la homografía.
7. Aplicar la homografía.

Ejemplo

- Alinear el formato escaneada con el formato original.



MAIL TO:
DMV CHANGE OF ADDRESS
P.O. BOX 98589
SACRAMENTO, CA 94209-0091

A SEPARATE FORM IS NEEDED FOR EACH DRIVER OR VEHICLE OWNER
NOTICE OF CHANGE OF ADDRESS

DMV USE ONLY
DL address
issued by FO
24002

Please Print Characters in Capital Letters Using Black or Dark Blue Ink only.
1 2 3 4 5 6 7 8 9 0 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Personal Information
LAST NAME: MALICK
FIRST: SATYA
MIDDLE: SATYA
BIRTH DATE: 01 01 1970
DRIVER LICENSE/DL CARD NO.: D1234567

Your Change of Address
We will change your voting address. If you are not registered to vote, you must complete a voter registration card. DMV provides the form or call 1-800-345-VOTE or login to the Secretary of State's website at www.sos.ca.gov.

Use only with DL Change of Address
 Mark this box if you do not want to change your voting address.

New or Correct Residence Address
STREET NUMBER ONLY: 10101
STREET NAME: BARRY STREET
APT. NO.:
CITY: DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME STATE: CA ZIP CODE: 10101

New or Correct Mailing Address
STREET NUMBER ONLY: 256
P.O. BOX OR STREET NAME OR STREET NAME AND PRIVATE MAIL BOX: ARTIFICIAL INTELLIGENCE STREET
APT. NO.:
CITY: DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME STATE: CA ZIP CODE: 12345

Vehicles, Vessels, or Pleasure Owned By You
Use Additional Forms if Necessary

Leased Vehicles
LEASE COMPANY'S NAME

Location of Trailer, Canoe, or Vessel
STREET NUMBER: STREET NAME:
CITY: DO NOT ABBREVIATE - USE FIRST 16 CHARACTERS IN CITY NAME COUNTY: DO NOT ABBREVIATE

Old Address
STREET NUMBER: STREET NAME: CITY: STATE: ZIP CODE:
Your mailing address may be given to requesters providing a valid reason for requesting the information. If you receive mail at your residence, then giving DMV a separate mailing address is optional for mail-order registrars per Vehicle Code Section 1808.21. Limit the parties whose names appear on the records) above and the mailing address shown is valid, existing and accurate. Consent to receive service of process at this mailing address pursuant to 416.2026, 416.30, and 416.30 of the CIV Procedure Code. I certify (or declare) under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

DMV 14 (REV. 03/07) 88888

Ejemplo

- Resultado.

DMV MAIL TO: DMV CHANGE OF ADDRESS, P.O. BOX 84389, SACRAMENTO, CA 95829-0081

A SEPARATE FORM IS NEEDED FOR EACH DRIVER OR VEHICLE OWNER

DMV USE ONLY
DL ADDRESS updated by FD 24002

NOTICE OF CHANGE OF ADDRESS

Please Print Characters in Capital Letters Using Black or Dark Blue Ink only.

INSTRUCTIONS: * Circle the mailing DMV records section (unavailable information cannot be updated). * Type or write your new address on a small piece of paper with your signature and date. Keep it with your driver license or ID card. * A commercial license driver must maintain a California residence address or the driver license will be downgraded to non-commercial status.

Personal Information
LAST NAME: MALLICK DRIVER LICENSE/ID CARD NO.: D1234567
FIRST: SATYA INITIAL: S BIRTH DATE: 01 01 1990

Voter Change of Address
We will change your voting address. If you are not registered to vote, you must complete a voter registration card. DMV provides the form or call 1-800-345-VOTE or login to the Secretary of State's website at www.sos.ca.gov.

DL Change of Address
Use only with DL Change of Address. Mark this box if you do not want to change your voting address.

New or Correct Residence Address
STREET NUMBER ONLY: 10101 STREET NAME: BINARY STREET
APT. NO.:
CITY - DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME: STATE: CA ZIP CODE: 10101

New or Correct Mailing Address
STREET NUMBER ONLY: 256 P.O. BOX OR STREET NAME OR STREET NAME AND PRIVATE MAIL BOX: ARTIFICIAL INTELLIGENCE
APT. NO.: STREET: STREET
CITY - DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME: STATE: CA ZIP CODE: 12345

Vehicles, Vessels, or Placards Owned By You
CALIFORNIA PLATE/PLACARD NO.: LAST 17 POSITIONS OF VEHICLE ID OR VESSEL HULL ID NUMBER: CHECK IF LEASED OR OUTSIDE CA:
LEARNOPENCV.COM

Leased Vehicles
LEASING COMPANY'S NAME:

Location of Trailer Coach or Vessel
STREET NUMBER: STREET NAME:
CITY - DO NOT ABBREVIATE - USE FIRST 18 CHARACTERS IN CITY NAME: COUNTY - DO NOT ABBREVIATE:

Old Address
STREET NUMBER/NAME: CITY: STATE: ZIP CODE:
Your mailing address may be given to requesters providing a valid reason for requesting the information. If you receive mail at your residence, then giving DMV a separate mailing address is optional. Your residence address is restricted to authorized requesters per Vehicle Code Section 18032.1. I am the person whose name appears on the record(s) above and the mailing address shown is valid, existing and accurate. I consent to receive service of process at this mailing address pursuant to 415.20(b), 415.30, and 416.60 of the Civil Procedure Code. I certify (or declare) under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

SIGNATURE: X

DMV 14 (REV. 5/01) WWW

Implementación

im1 = imagen que se quiere alinear

im2 = imagen de referencia

Detect ORB features and compute descriptors

```
orb = cv2.ORB.create()
```

```
kp1, desc1 = orb.detectAndCompute(im1, None)
```

```
kp2, desc2 = orb.detectAndCompute(im2, None)
```

Match features

```
bf = cv2.BFMatcher.create(cv2.NORM_HAMMING, True)
```

```
matches = bf.match(desc1, desc2)
```

Sort matches by distance

```
matches = sorted(matches, key=lambda x: x.distance, reverse=False)
```

Implementación

Remove not so good matches

```
numGoodMatches = int(len(matches) * GOOD_MATCH_PERCENT)
matches = matches[:numGoodMatches]
```

Extract location of good matches

```
points1 = np.zeros((len(matches), 2), dtype=np.float32)
points2 = np.zeros((len(matches), 2), dtype=np.float32)
for i, match in enumerate(matches):
    points1[i, :] = keypoints1[match.queryIdx].pt
    points2[i, :] = keypoints2[match.trainIdx].pt
```

Implementación

Find homography

```
h, mask = cv2.findHomography(points1, points2, cv2.RANSAC)
```

Use homography

```
height, width, channels = im2.shape
```

```
im_reg = cv2.warpPerspective(im1, h, (width, height))
```