

▼ Image Alignment

Satya Mallick, LearnOpenCV.com

Align an image to a template.



Theory

scanned-form.jpg form.jpg X ...

DMV
A Public Service Agency

MAIL TO:
DMV CHANGE OF ADDRESS
P O BOX 94259
SACRAMENTO, CA 94259-0001

**A SEPARATE FORM IS NEEDED
FOR EACH DRIVER OR VEHICLE OWNER**

NOTICE OF CHANGE OF ADDRESS

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

INSTRUCTIONS:

- Enter the information as shown on the document, i.e. California driver license, ID card, or vehicle registration card, for which a change is being requested.
- Or enter the information as shown on the document, i.e. California driver license, ID card, or vehicle registration card, for which a change is being requested.
- Type or write your new address on a separate sheet of paper with your signature and date. Keep it with your driver license or ID card.
- A commercial licensed driver must maintain a California residence address or the driver license will be downgraded to non-commercial status.

Personal Information

LAST NAME	DRIVER LICENSE/CARD NO.
FIRST	INITIAL
MM-DD-YYYY	

Voter Change of Address

Use only with DL Change of Address

New or Correct Residence Address

Do Not Use P.O. Box in this space

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 logon to the Secretary of State's website at www.sos.ca.gov.

New or Correct Mailing Address

If Different From Residence Address

Vehicles, Vessels, or Placards Owned By You

Leased Vehicles

Location of Trailer Coach or Vessel

Old Address

STREET NUMBER ONLY STREET NAME

APT NO.

CITY - DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME STATE ZIP CODE

STREET NUMBER ONLY P. O. BOX OR STREET NAME OR STREET NAME AND PRIVATE MAIL BOX

APT NO.

CITY - DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME STATE ZIP CODE

CALIFORNIA PLATE/CF/PLACARD NO. LAST 17 POSITIONS OF VEHICLE ID OR VESSEL HULL ID NUMBER

LEASED COMPANY'S NAME

STREET NUMBER STREET NAME

CITY - DO NOT ABBREVIATE - USE FIRST 16 CHARACTERS IN CITY NAME COUNTY - DO NOT ABBREVIATE

STREET NUMBERNAME CITY STATE ZIP CODE

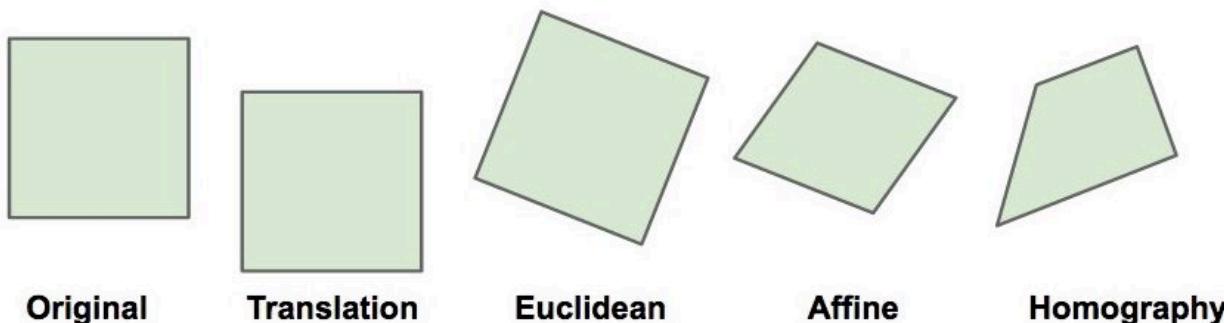
Your mailing address may be given to requesters providing a valid reason for requesting the information. If you receive mail at your residence, then giving DMV a separate mailing address is optional. Your residence address is restricted to authorized requesters per Vehicle Code Section 1808.21. I am 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 416.90 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

MM-DD-YYYY

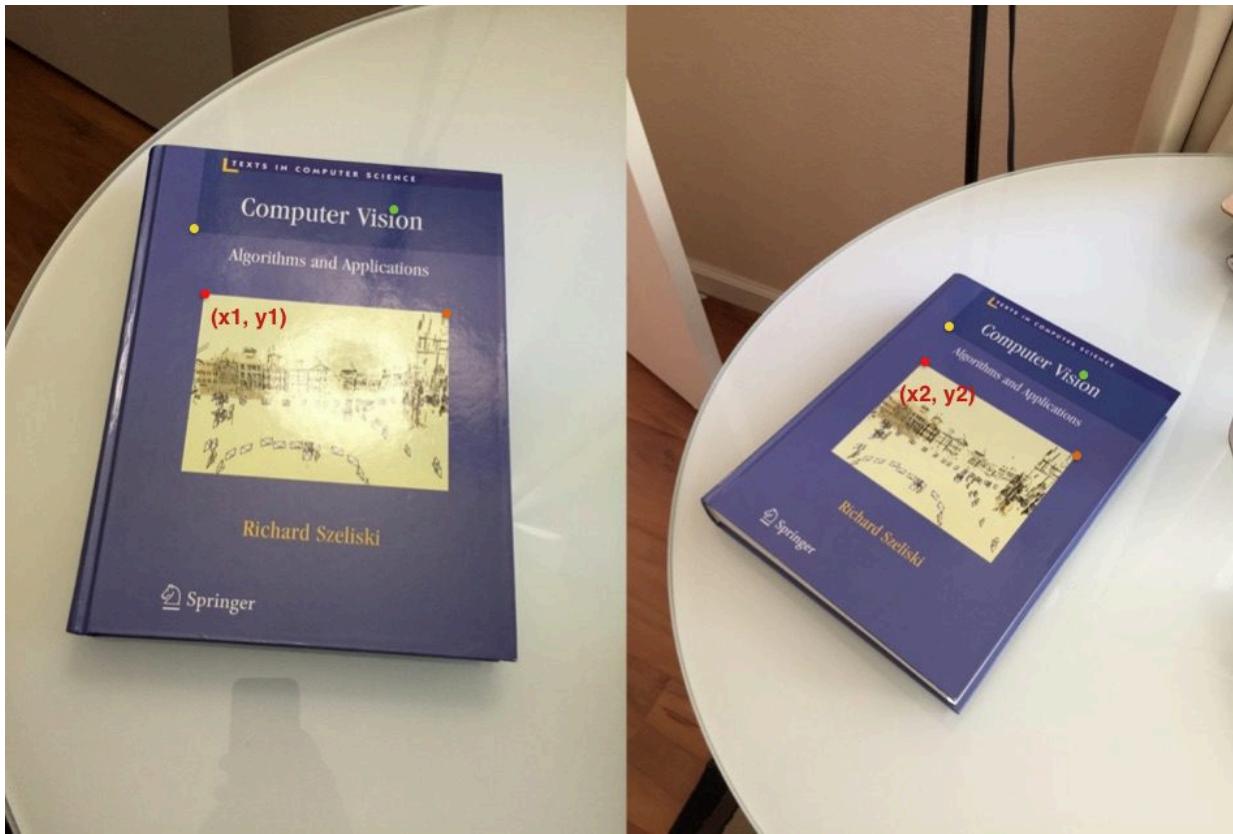
Print Clear Form

1. A **Homography** transforms a square to arbitrary quad.



▼ Theory

2. Images of two planes are related by a **Homography**
3. We need **4 corresponding points** to estimate Homography



```
# Imports
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt

from zipfile import ZipFile
from urllib.request import urlretrieve

%matplotlib inline
```

▼ Download Assets

```
def download_and_unzip(url, save_path):
    print(f"Downloading and extracting assets....", end="")

    # Downloading zip file using urllib package.
    urlretrieve(url, save_path)

    try:
        # Extracting zip file using the zipfile package.
        with ZipFile(save_path) as z:
            # Extract ZIP file contents in the same directory.
            z.extractall(os.path.split(save_path)[0])

        print("Done")

    except Exception as e:
        print("\nInvalid file.", e)

URL = r"https://www.dropbox.com/s/zuwnn6rqe0f4zgh/opencv_bootcamp_assets_NB8.zip?

asset_zip_path = os.path.join(os.getcwd(), "opencv_bootcamp_assets_NB8.zip")

# Download if asset ZIP does not exists.
if not os.path.exists(asset_zip_path):
    download_and_unzip(URL, asset_zip_path)
```

→ Downloading and extracting assets....Done

▼ Step 1: Read Tempalate and Scanned Image

```
# Read reference image
refFilename = "form.jpg"
print("Reading reference image:", refFilename)
im1 = cv2.imread(refFilename, cv2.IMREAD_COLOR)
im1 = cv2.cvtColor(im1, cv2.COLOR_BGR2RGB)

# Read image to be aligned
imFilename = "scanned-form.jpg"
print("Reading image to align:", imFilename)
im2 = cv2.imread(imFilename, cv2.IMREAD_COLOR)
im2 = cv2.cvtColor(im2, cv2.COLOR_BGR2RGB)
```

→ Reading reference image: form.jpg
Reading image to align: scanned-form.jpg

```
# Display Images
```

```
plt.figure(figsize=[20, 10]);
plt.subplot(121); plt.axis('off'); plt.imshow(im1); plt.title("Original Form")
plt.subplot(122); plt.axis('off'); plt.imshow(im2); plt.title("Scanned Form")
```

→ Text(0.5, 1.0, 'Scanned Form')

Original Form

DMV MAIL TO: DMV CHANGE OF ADDRESS
J. P. LEE, DIRECTOR SACRAMENTO, CA 94291-0001

A SEPARATE FORM IS NEEDED FOR EACH DRIVER OR VEHICLE OWNER

NOTICE OF CHANGE OF ADDRESS

24002

INSTRUCTIONS: Complete this form in ink. Use black or dark blue ink only.
 • Please do not use ink pens, markers, or felt tip pens.
 • Please do not use ink pens, markers, or felt tip pens.
 • Please do not use ink pens, markers, or felt tip pens.
 • Please do not use ink pens, markers, or felt tip pens.

Personal Information

LAST NAME: [REDACTED] FIRST NAME: [REDACTED] MIDDLE NAME: [REDACTED]

BIRTH DATE: [REDACTED]

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 1-800-34-VOTE or log on to the Secretary of State's website at www.sos.ca.gov.

Mark this box if you do not want to change your voting address.

New or Correct Mailing Address

STREET NUMBER ONLY: [REDACTED] STREET NAME: [REDACTED]

APT NO.: [REDACTED]

CITY - DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME: [REDACTED] STATE: [REDACTED] ZIP CODE: [REDACTED]

New or Correct Mailing Address

STREET NUMBER ONLY: [REDACTED] P.O. BOX OR STREET NAME OR STREET NAME AND PRIVATE MAIL BOX: [REDACTED]

APT NO.: [REDACTED]

CITY - DO NOT ABBREVIATE - USE FIRST 22 CHARACTERS IN CITY NAME: [REDACTED] STATE: [REDACTED] ZIP CODE: [REDACTED]

Vehicles, Vessels, or Property Owned By You

CALIFORNIA PLATES/PLACARD NO.: [REDACTED] LAST 17 POSITION OF VEHICLE ID OR VESSEL/HULL ID NUMBER: [REDACTED]

CHECK IF CHECKED
LEASER OUTSIDE CA

Leased Vehicles

LEASING COMPANY'S NAME: [REDACTED]

Location of Trailer/Coach/Vessel

STREET NUMBER: [REDACTED] STREET NAME: [REDACTED]

Old Address

STREET NUMBER: [REDACTED] STREET NAME: [REDACTED]

STATE: [REDACTED] COUNTY: [REDACTED]

Signature: [REDACTED]

DMV 14 (REV. 5/2017) WWW

Print Clear Form

Scanned Form



▼ Step 2: Find keypoints in both Images

Think of keypoints as corner points that are stable under image transformations

```
# Convert images to grayscale
im1_gray = cv2.cvtColor(im1, cv2.COLOR_BGR2GRAY)
im2_gray = cv2.cvtColor(im2, cv2.COLOR_BGR2GRAY)

# Detect ORB features and compute descriptors.
MAX_NUM_FEATURES = 500
orb = cv2.ORB_create(MAX_NUM_FEATURES)
keypoints1, descriptors1 = orb.detectAndCompute(im1_gray, None)
keypoints2, descriptors2 = orb.detectAndCompute(im2_gray, None)

# Display
im1_display = cv2.drawKeypoints(im1, keypoints1, outImage=np.array([]),
                                 color=(255, 0, 0), flags=cv2.DRAW_MATCHES_FLAGS_D

im2_display = cv2.drawKeypoints(im2, keypoints2, outImage=np.array([]),
                                 color=(255, 0, 0), flags=cv2.DRAW_MATCHES_FLAGS_D

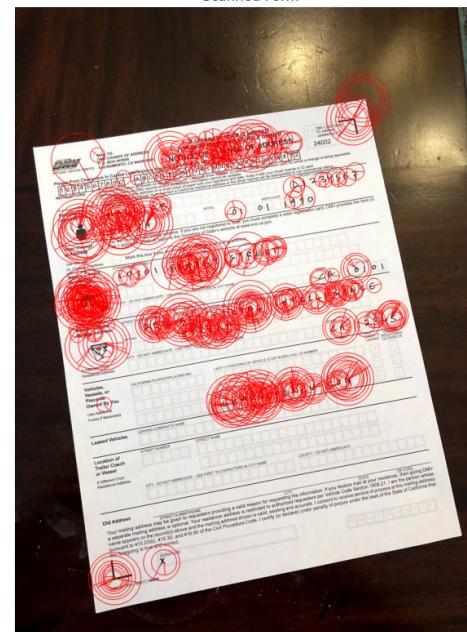
plt.figure(figsize=[20,10])
plt.subplot(121); plt.axis('off'); plt.imshow(im1_display); plt.title("Original F
plt.subplot(122); plt.axis('off'); plt.imshow(im2_display); plt.title("Scanned Fo
```



Original Form

This image shows the original DMV form with numerous red circles drawn over it, highlighting specific fields and sections for analysis. The highlighted areas include the header, address sections, vehicle information, and signature areas.

Scanned Form



Step 3 : Match keypoints in the two image

```
# Match features.
matcher = cv2.DescriptorMatcher_create(cv2.DESCRIPTOR_MATCHER_BRUTEFORCE_HAMMING)

# Converting to list for sorting as tuples are immutable objects.
matches = list(matcher.match(descriptors1, descriptors2, None))

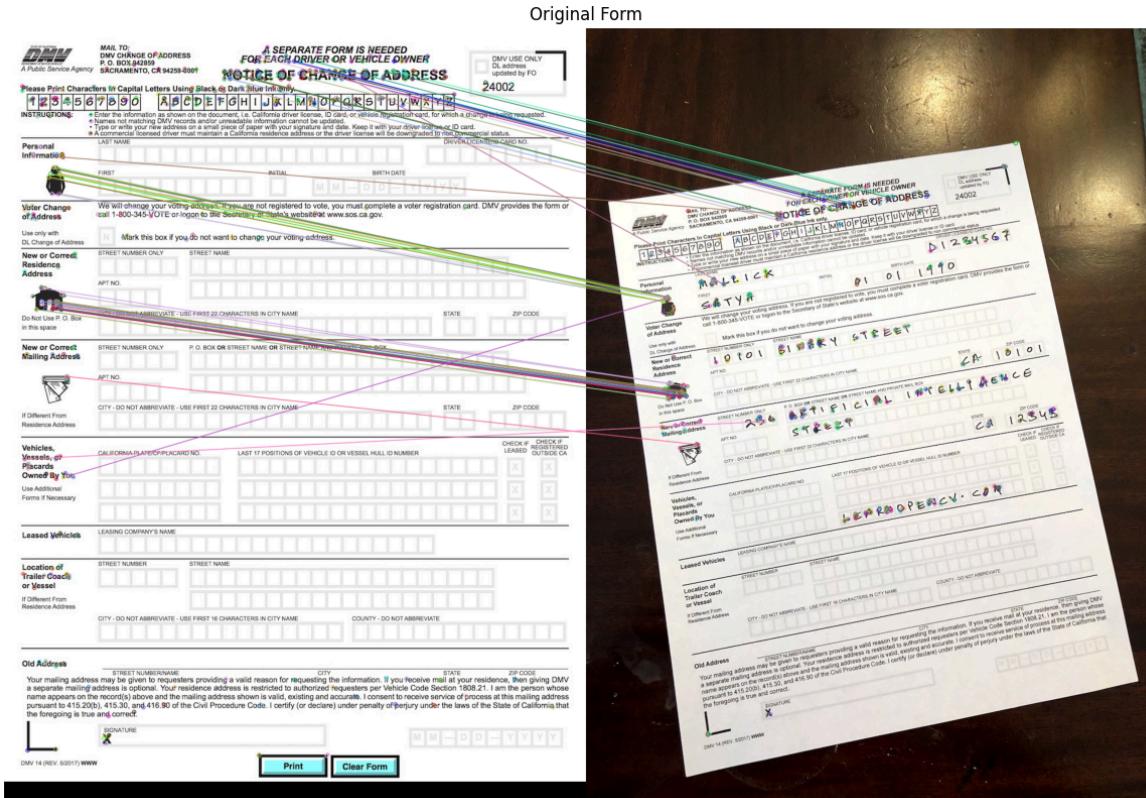
# Sort matches by score
matches.sort(key=lambda x: x.distance, reverse=False)

# Remove not so good matches
numGoodMatches = int(len(matches) * 0.1)
matches = matches[:numGoodMatches]
```

```
# Draw top matches
im_matches = cv2.drawMatches(im1, keypoints1, im2, keypoints2, matches, None)

plt.figure(figsize=[40, 10])
plt.imshow(im_matches);plt.axis("off");plt.title("Original Form")
```

→ Text(0.5, 1.0, 'Original Form')



▼ Step 4: Find Homography

```
# Extract location of good matches
```