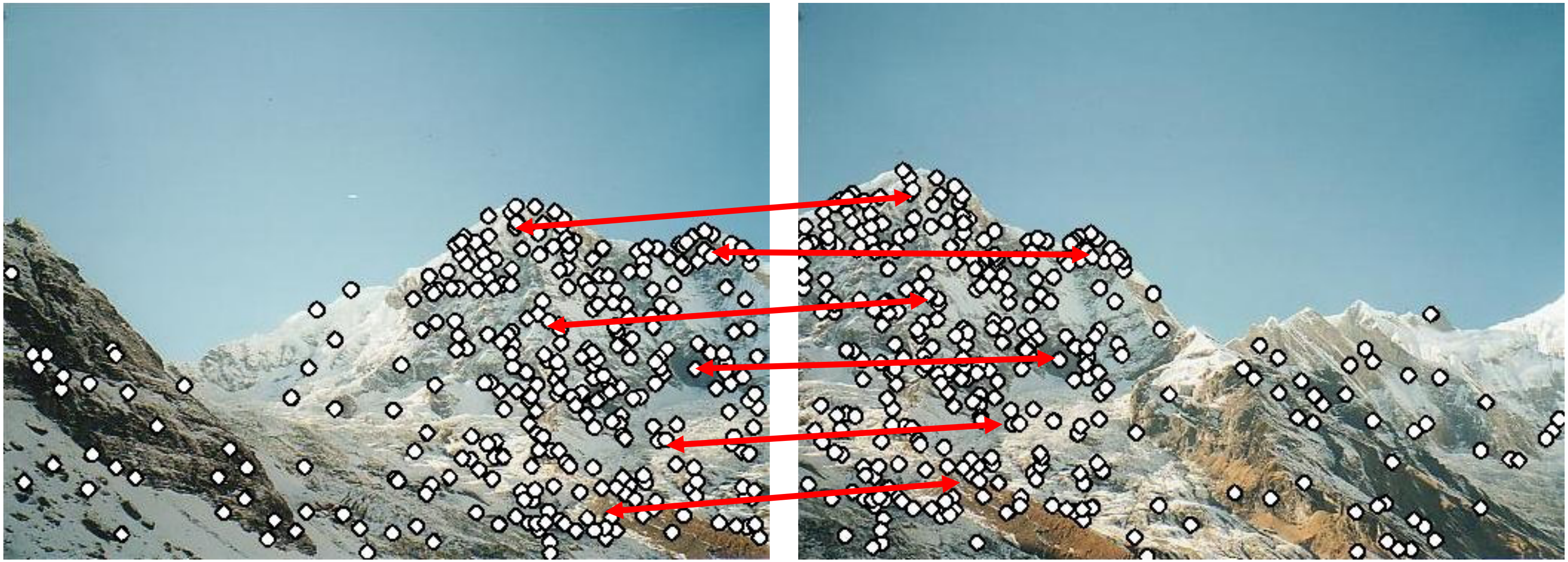
# CSCI 497P/597P: Computer Vision



## Lecture 11: Image Features

Feature Matching

Forward and Inverse Warping

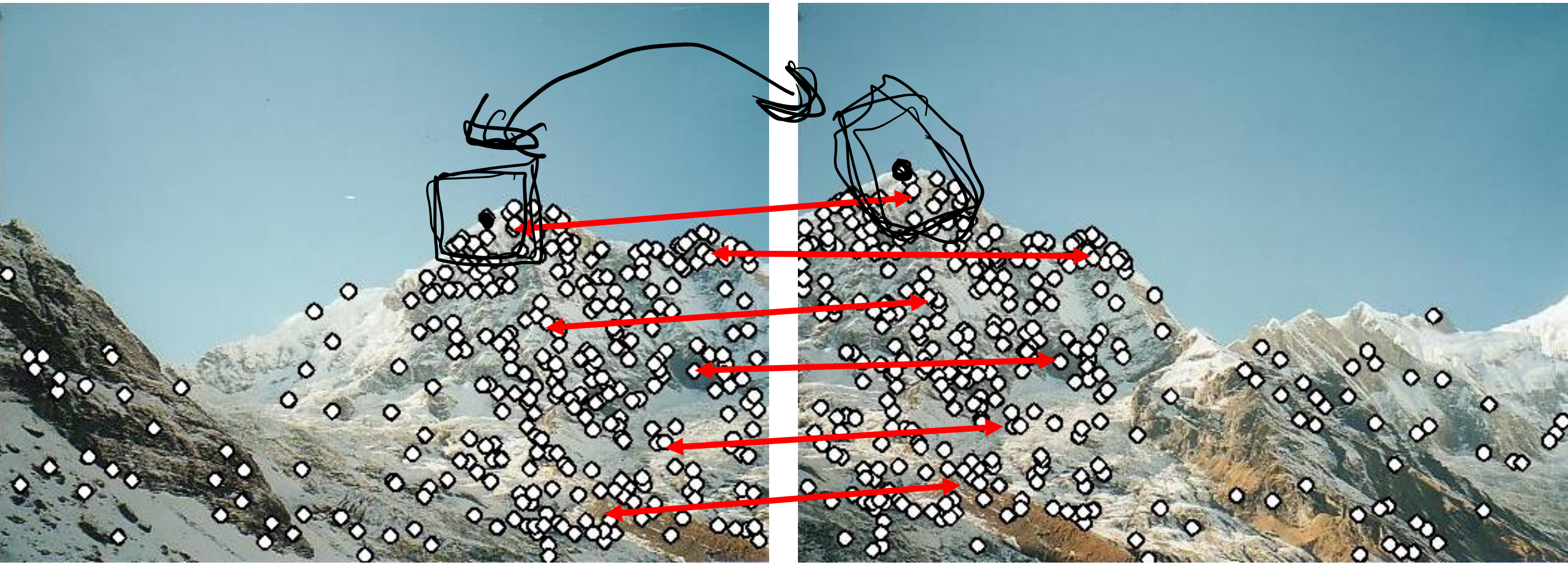
# Announcements

- Feedback survey
- HW 1 is due Wednesday

# Goals

- Know how and why to **match** features using:
  - The simple SSD metric
  - The ratio test
- Understand the mathematical framework for (lineare) geometric transformations on images (image warping).
- Understand the differences between forward and inverse warping.

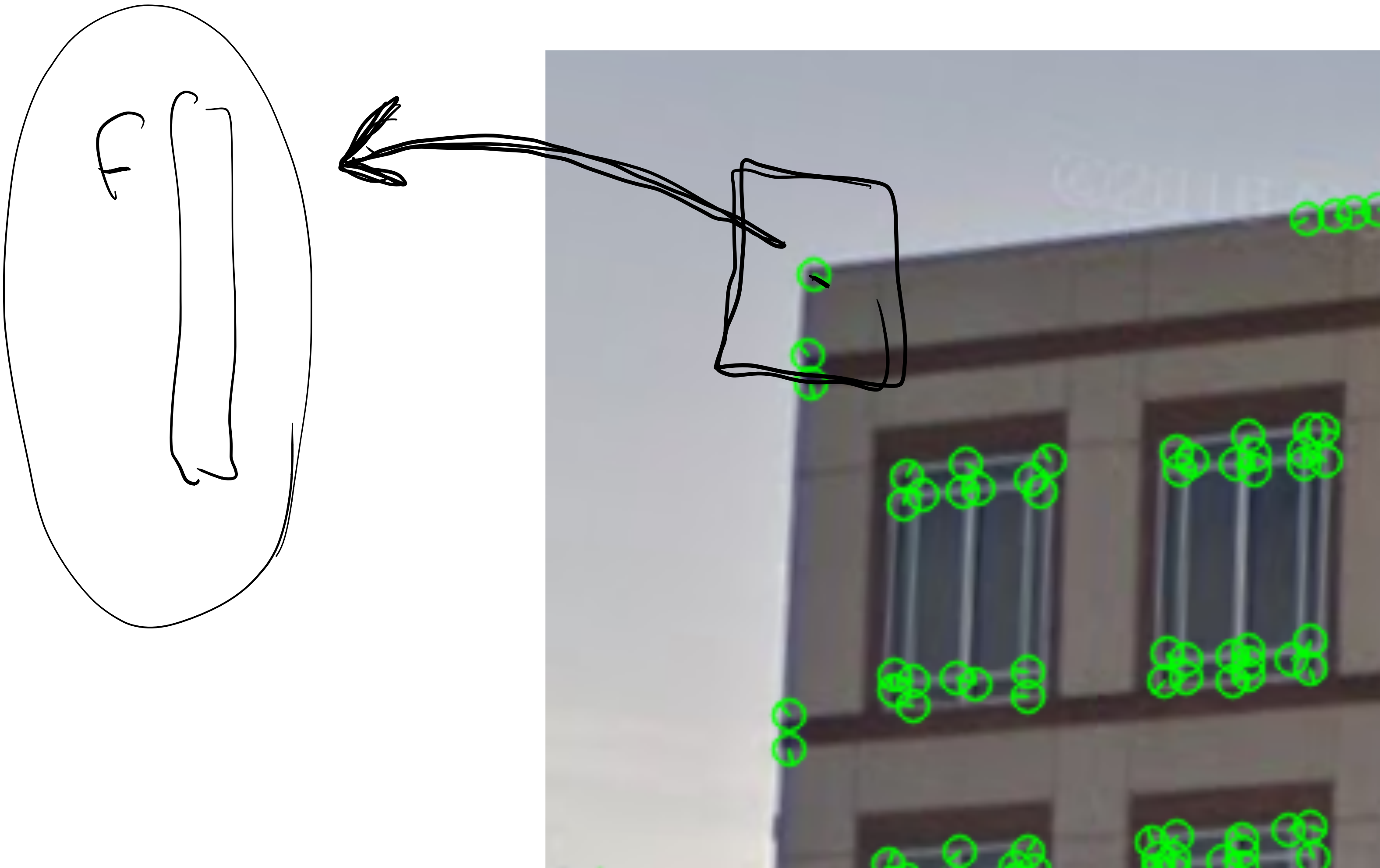
# Running motivational example: Panorama Stitching



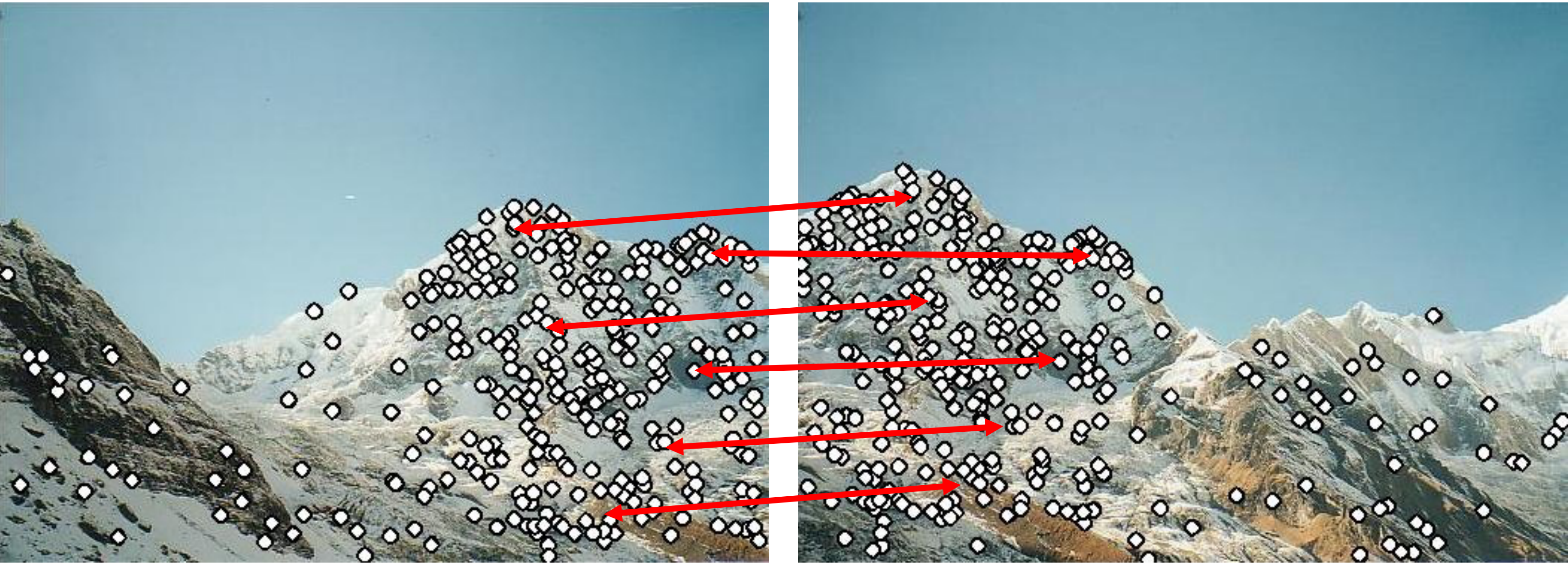
# 1. Detect Harris Corners

# 1. Detect Harris Corners

# 2. Describe features



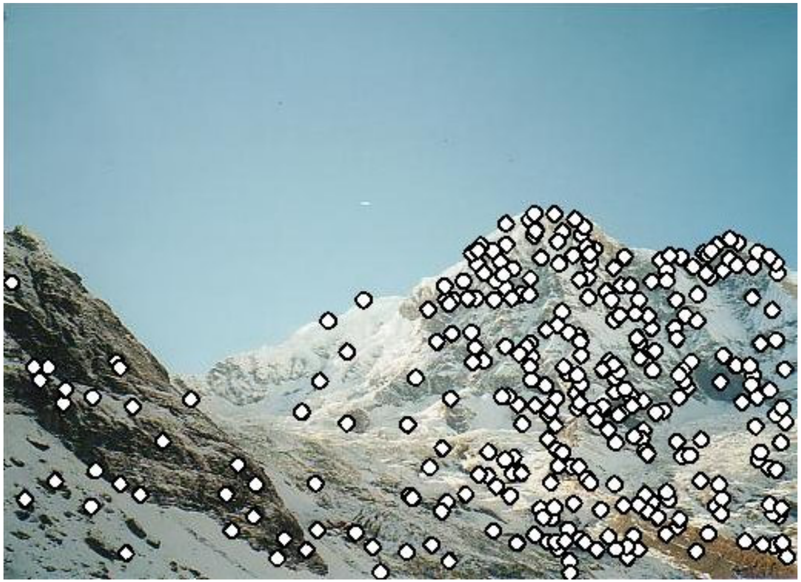
# 3. Match Features



At this point, a "feature" has:

1. A keypoint: its position in one of the images
2. A descriptor: a vector of numbers that 'captures' its characteristics

# Feature Matching



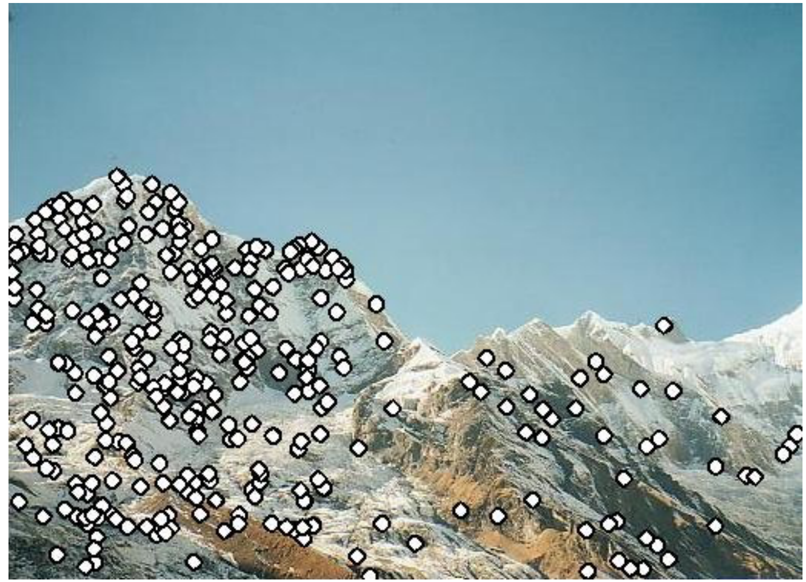


Image 1 features:

$F1 = \{ f_1^1, f_2^1, \ldots \}$

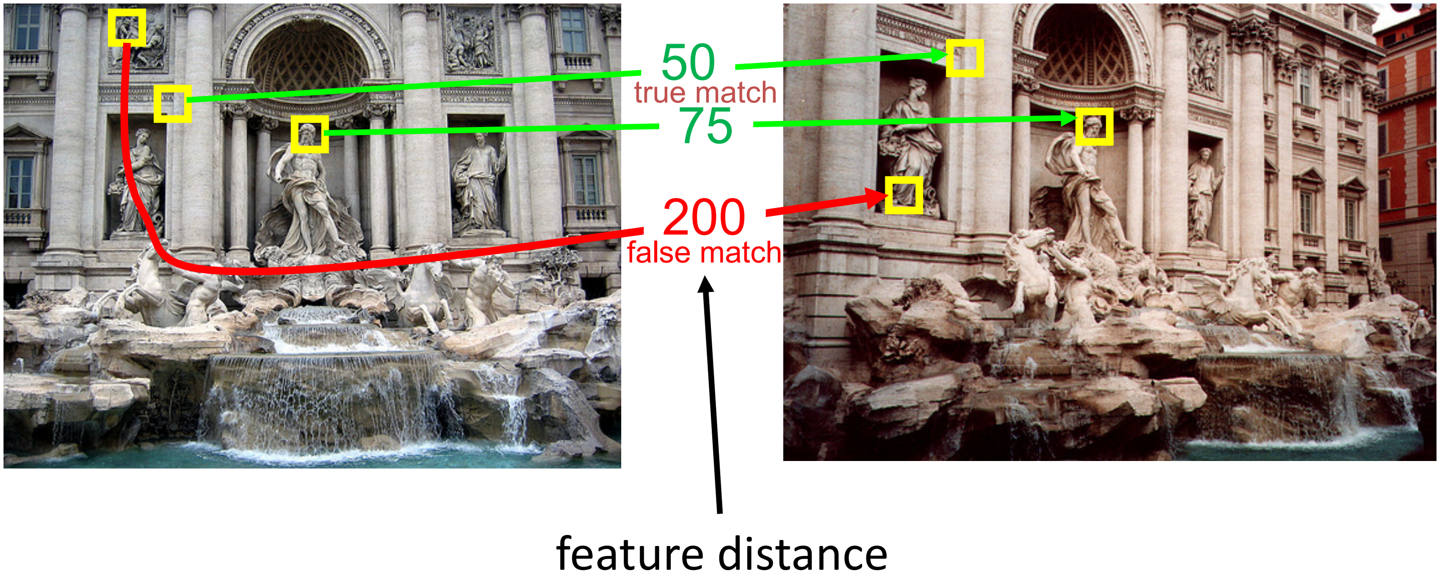
Image 2 features:

$F2 = \{ f_1^2, f_2^2, \ldots \}$

# Matching Algorithm

```
F1 = detect_describe(img1)

F2 = detect_describe(img2)

for f1 in F1:

    find f2 that minimizes d(f1, f2)

    add (f1, f2) to matches
```

# But what if we're wrong?




- Answer #1: Threshold on match score

# Matching Algorithm

```
F1 = detect_describe(img1)

F2 = detect_describe(img2)

for f1 in F1:

    find f2 that minimizes d(f1, f2)

    if d(f1, f2) < T  ← choose empirically

        add (f1, f2) to matches
```

# Distance Metrics

What should we use for **d** in **d**(f1, f2)? SSD

Sum of Squared Differences

$$\|f_1 - f_2\|^2 = \sum \left( f_1 - f_2 \right)^2$$

# Distance Metrics

Sidenote: efficiently computing SSD

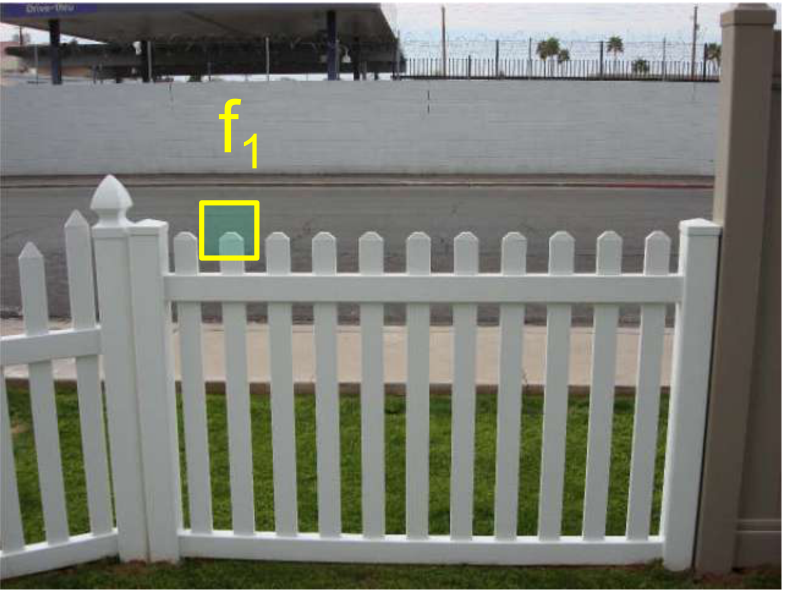
```
np.sum((f1 - f2)**2)
```

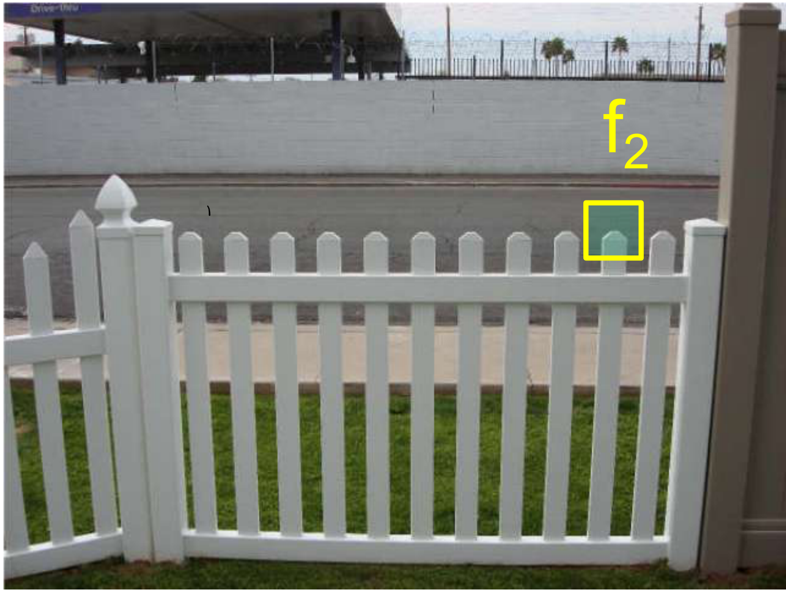
```
d = f1 - f2
dist = d.dot(d)
```

$d^T$ $d$

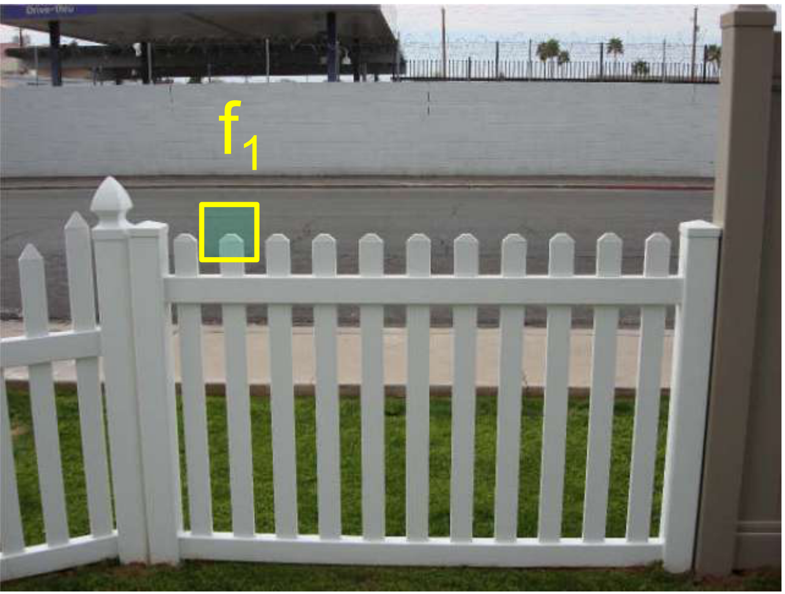
# A problem with SSD

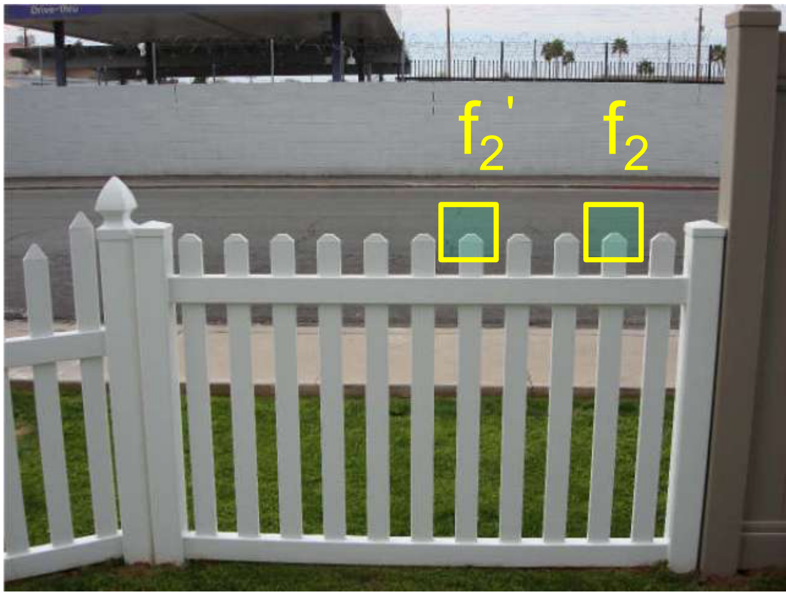
# A problem with SSD



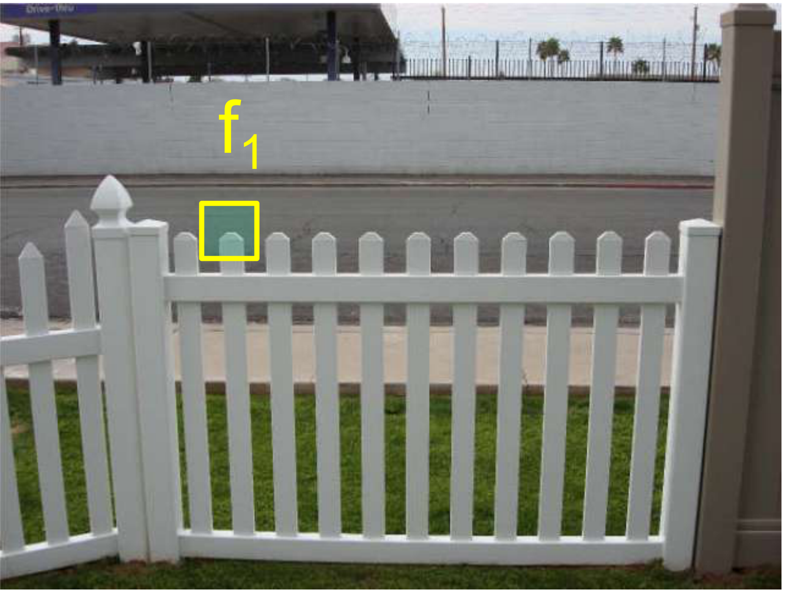



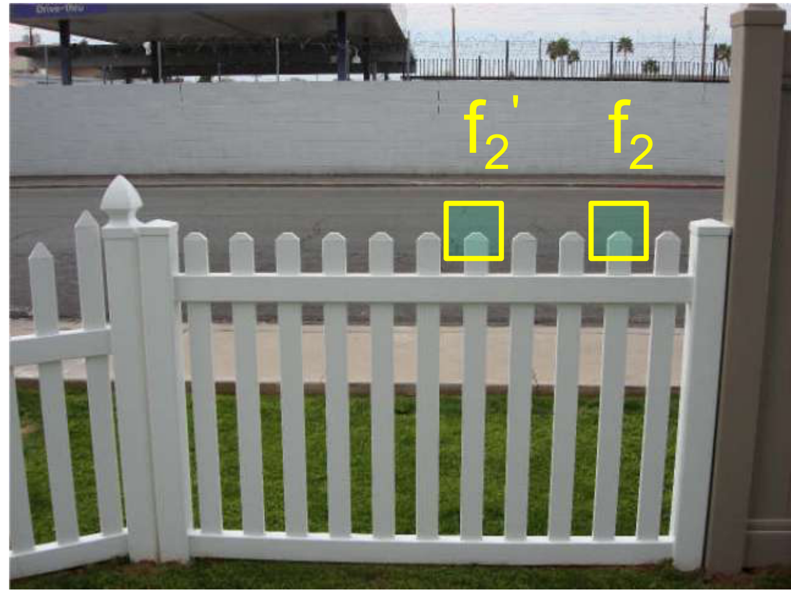
# A problem with SSD



# Ratio test - Intuition




We want a metric that gives small distance when features are

- like each other (according to SSD), but
- not like any others

# "Ratio Test" distance metric

Small

$$r = \frac{\|f_1 - f_2\|^2}{\|f_1 - f_2'\|^2}$$

large

$f_2$ = closest SSD

$f_2'$ = second closest

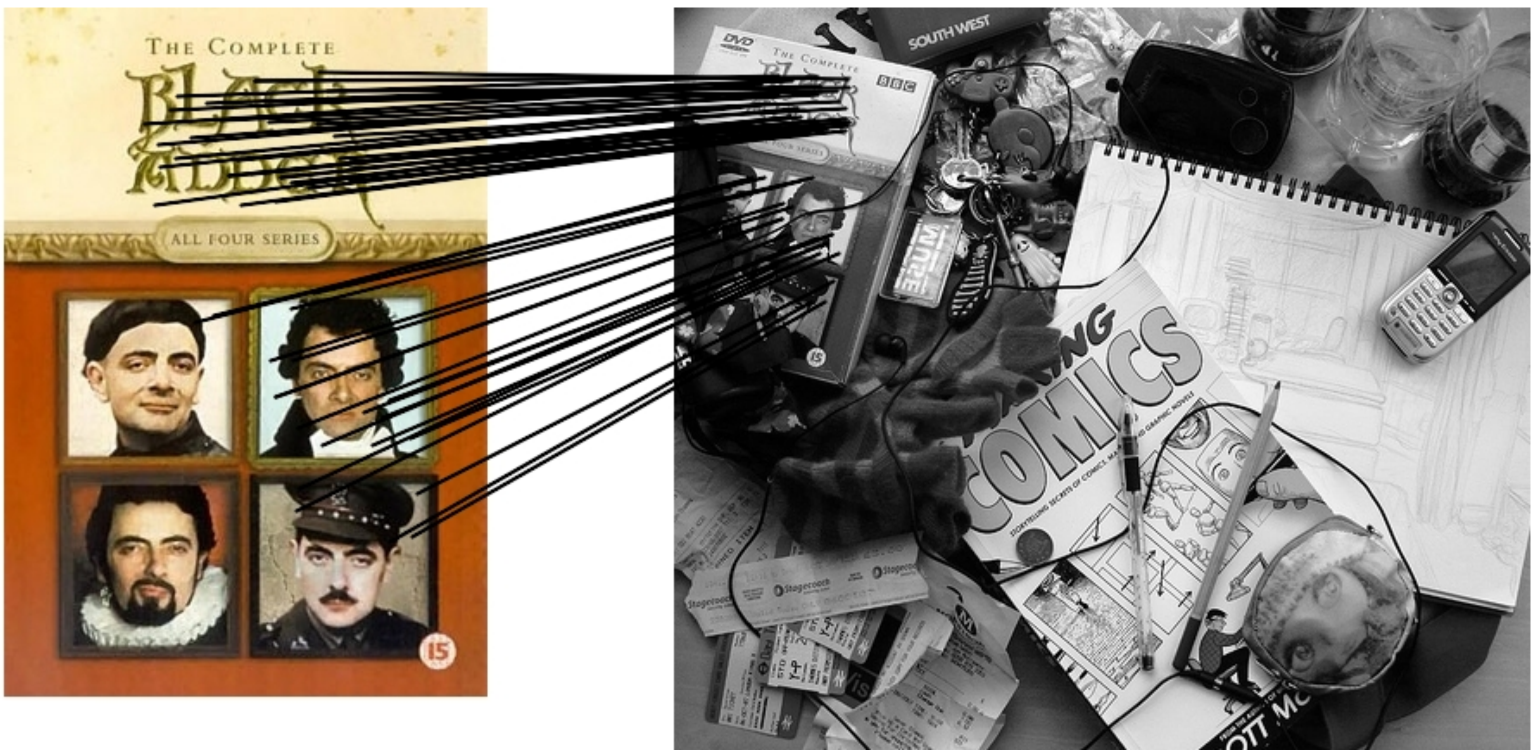
f1's closest match is still f2, but the distance between them is different

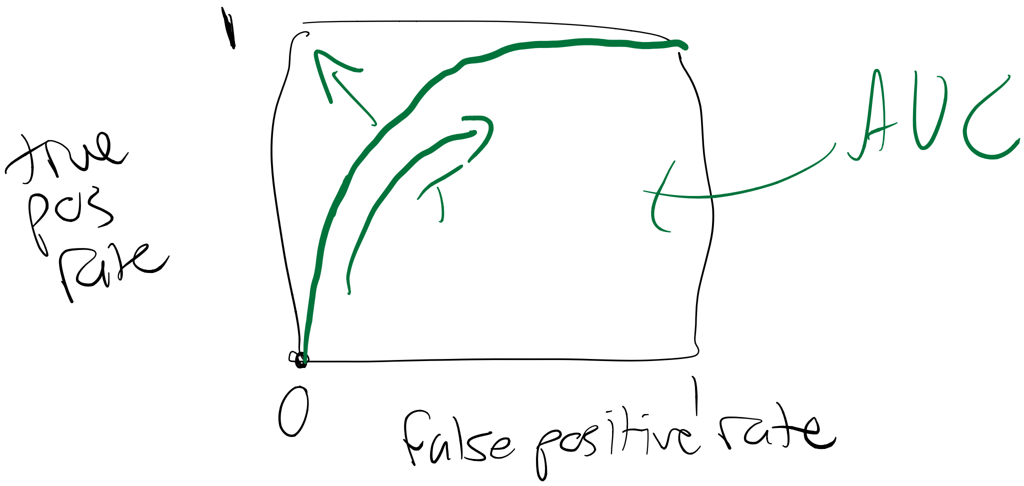
# Matching Algorithm

```
F1 = detect_describe(img1)

F2 = detect_describe(img2)

for f1 in F1:

   f2 = closest match according to SSD

   f2' = second-closest match according to SSD

   if SSD(f1,f2) / SSD(f1, f2') < T

       add (f1, f2) to matches
```

# But what if we're wrong?

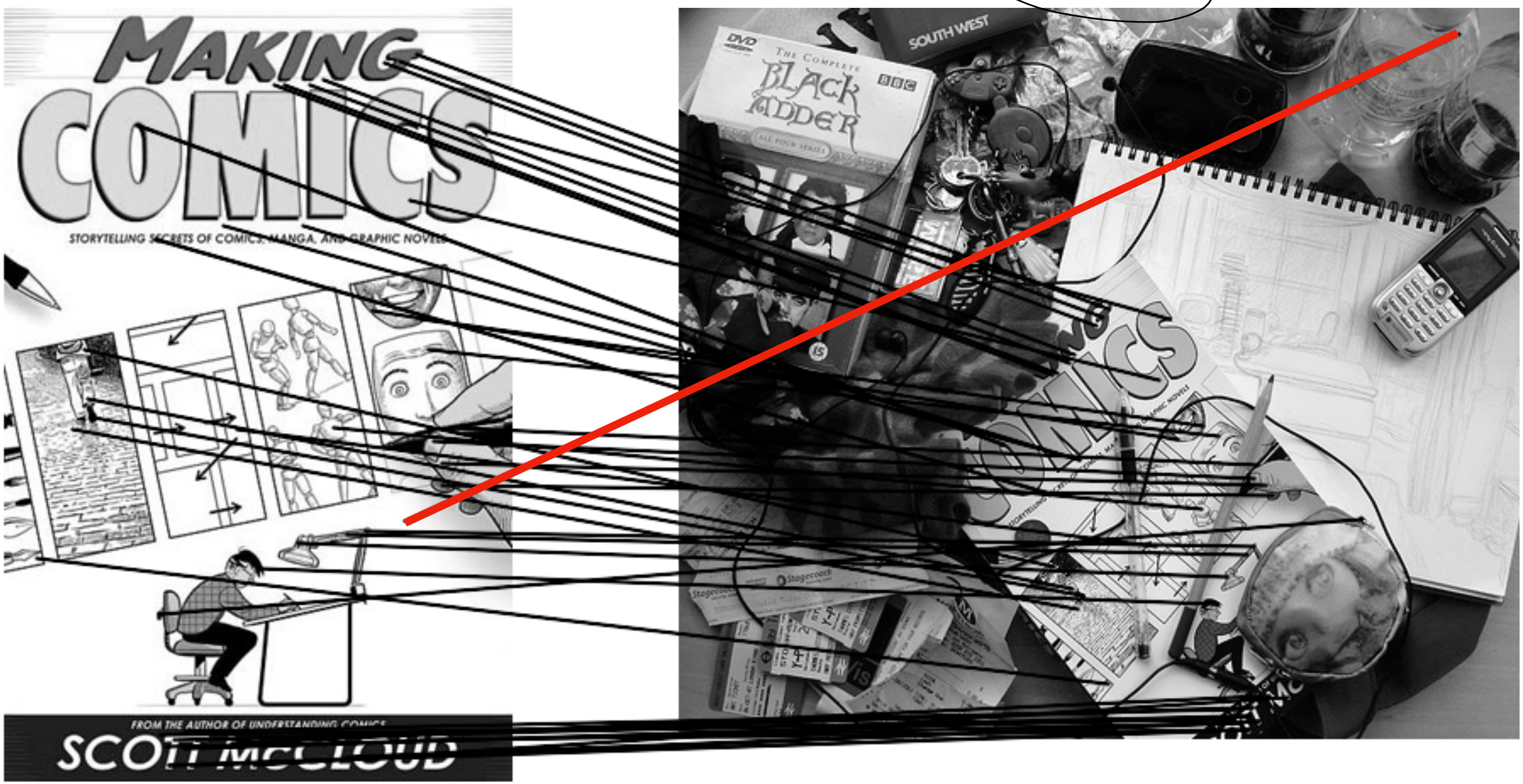
Decreasing the threshold T gives fewer **false positives** but also fewer **true positives**.




1
true pos rate
0
false positive rate
1
AUC

# But what if we're wrong?

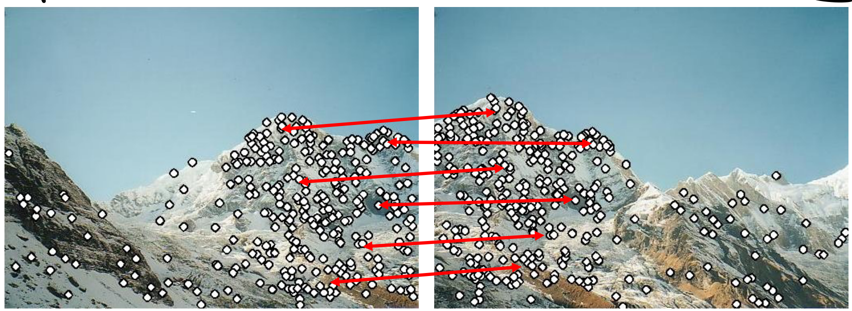
Many good matches, but still a few **outliers**.



# Let's suppose we were never wrong.

We have a **perfect** set of feature matches.

$I_1$ Great job team! $I_2$



1. Find $T$

2. Warp $I_2$ into $I_1$'s coords

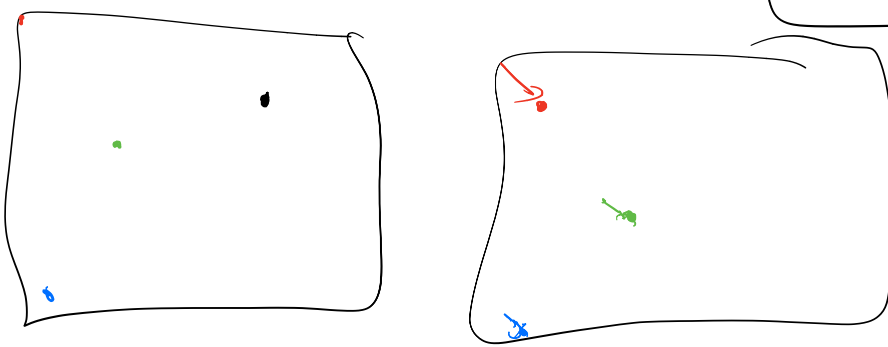
# Suppose

Match keypoints:

| Img1 | Img2 |
|---|---|
| x y | x y |
| (0,0) | (2,2) |
| (74, 68) | (76, 70) |
| (104, 6), | (106, 8) |

$x', y' = f(x, y)$

$$\boxed{\begin{aligned} x' &= x + 2 \\ y' &= y + 2 \end{aligned}}$$



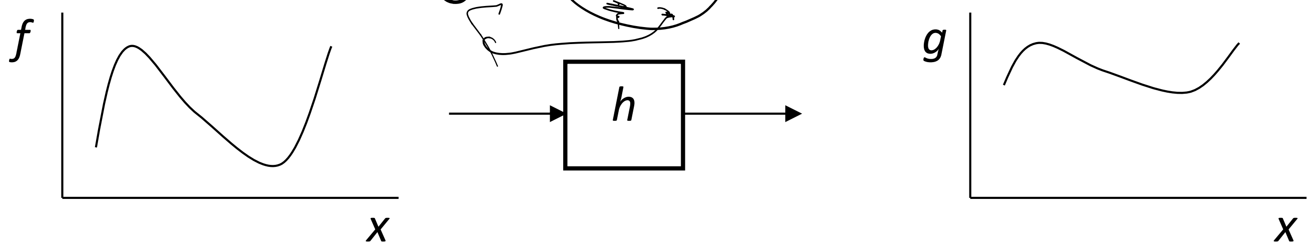
# This is a geometric transformation

filter

$(\mathbb{R}^2 \to \mathbb{R}) \to (\mathbb{R}^2 \to \mathbb{R})$

- **Filtering** was a transformation on the *range:*

$$g(x) = h(f(x))$$

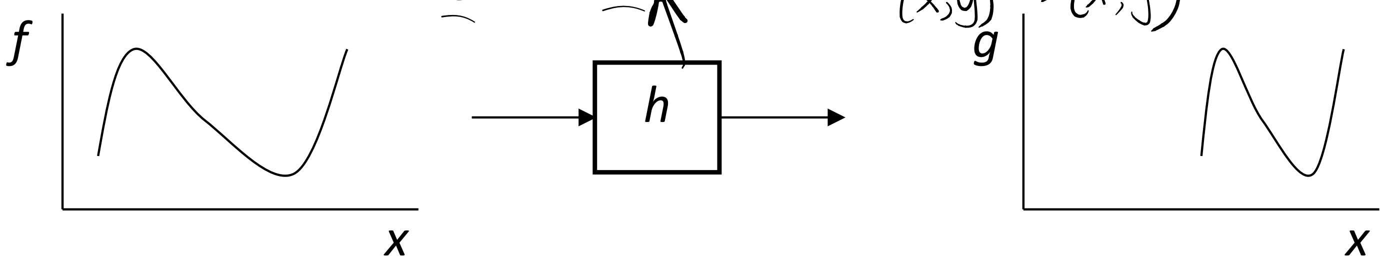



- **Warping** is a transformation on the domain:

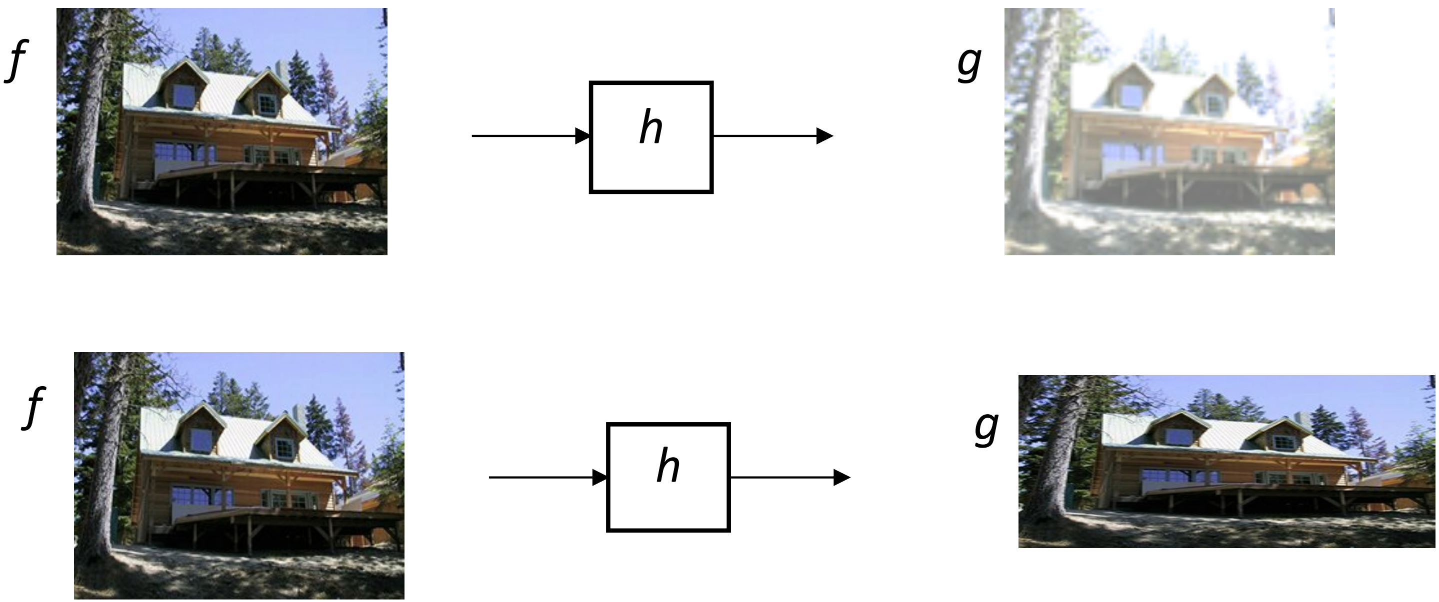
$$g(x) = f(h(x))$$

$h : \mathbb{R}^2 \to \mathbb{R}^2$

$(x, y) \to (x', y')$



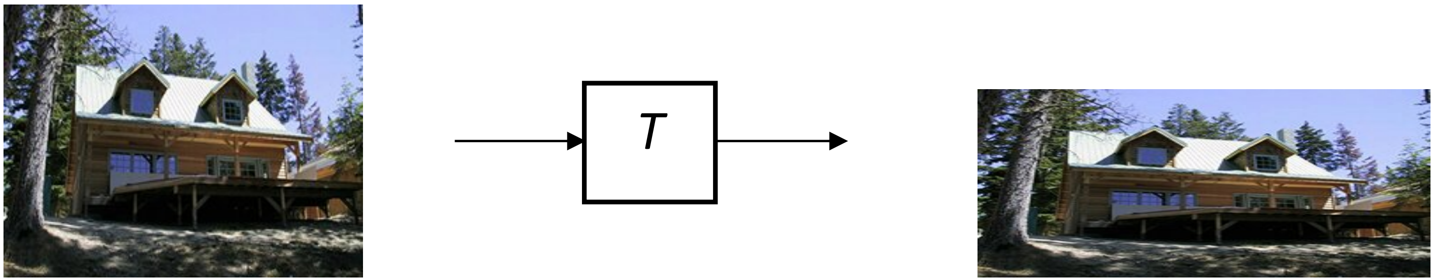
# Filtering vs Warping



Linear

# ~~Parametric~~ (global) Warping

- Apply the same function to all coordinates.




**p** = (x,y)   T transforms *image coordinates*   **p'** = (x',y')

x', y' = T(x, y)

# Suppose

Match keypoints:

| Img1 | Img2 |
|---|---|
| (1, 1) | (2,2) |
| (24, 63) | (48, 126) |
| (10, 900), | (20, 1800) |

x', y' = T(x, y)

What is function T describes the transformation between these?

What is T?

$x' = 2x$

$y' = 2y$

# Linear Transformations

- Linear means matrices, right?

$$\text{Img2 Coords}\ \begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}}_{T} \begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \text{Img1 Coords}$$

# What can we do with these?

$$S = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

scale uniformly by $s$

$$S = \begin{bmatrix} S_x & 0 \\ 0 & S_y = 1 \end{bmatrix}$$