Lect 9 Image Alignment

Feature based alignment

1. Search for feature matches
   * Local appearance.
     feature shows up in both images
   * Global configuration
     location, relative geometric of feature is the same.

2. Treat alignment as fitting.

$$\min \sum_x \text{residual}(x, \text{Model}).$$

model for image transformation.

Image transformation models:

① Translation

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+tx \\ y+ty \\ 1 \end{bmatrix}$$

only 2 params, tx, ty,
preserve orientation
T(tx, ty) 2-DoF. called Translation

② Translation & Rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta & tx \\ \sin\theta & \sin\theta & ty \\ 0 & 0 & 1 \end{bmatrix}$$

+ rotation θ. by origion

③ Scaling

$$\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

+ 1 param.
(ignore isymmetric scaling)

If we combine $T(t_x, t_y, \theta, a)$. 4-DoF. angle preserved.
it's called <u>Similarity Transformation</u>

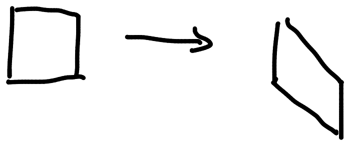
④ Aspect

$$\begin{bmatrix} a_a & & \\ & 1/a_a & \\ & & 1 \end{bmatrix}$$

+1 DoF

⑤ Shear

$$\begin{bmatrix} 1 & a_s & 0 \\ b_s & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



+1 DoF, because we can manually set $b_s = 1$ by rotation first

$T(t_x, t_y, \theta, a, a_a, a_s)$ 6-DoF, only preserve parallel.
called <u>Affine Transformation</u>

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \iff \underbrace{\begin{bmatrix} a & b \\ d & e \end{bmatrix}}_{M} \begin{bmatrix} X \\ Y \end{bmatrix} + \underbrace{\begin{bmatrix} c \\ f \end{bmatrix}}_{t}$$

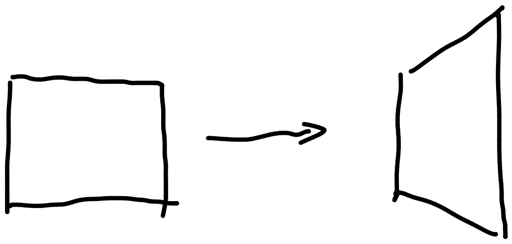
$$X' = MX + t$$

We can solve it by least-square.

$$\min_{M,t} \sum_i \| X_i' - MX_i + t \|_2^2 \iff \min_{M,t} \left\| \begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ b \\ d \\ e \\ c \\ f \end{bmatrix} - \begin{bmatrix} x_i' \\ y_i' \\ \vdots \\ \vdots \end{bmatrix} \right\|_2^2$$

Homography For non-orthography camera

It's a projection



because parallel may not be parallel

Pay attention!

Homography could only deal with rotation of camera,

$\boxed{\text{No! Movement!}}$ Because no one could predict things not shown on the picture.

Note that, For Homogenious coordinate, scaling doesn't care

$$\lambda \begin{bmatrix} a \\ b \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \\ w \end{bmatrix}$$

Now

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} H \end{bmatrix}_{3\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

H has 3 DoF, because scaling doesn't matter.

$\lambda X' = HX$ note that, λ can be 0 to represent infinite point

$$\lambda X' = HX \iff X' \parallel HX, \iff X' \times HX = 0$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \times \begin{bmatrix} h_1^T x \\ h_2^T x \\ h_3^T x \end{bmatrix} = \begin{bmatrix} y' h_3^T x - h_2^T x \\ h_1^T x - x' h_3^T x \\ x' h_2^T x - y h_1^T x \end{bmatrix}$$

$$\Rightarrow \underbrace{\begin{bmatrix} 0 & X_i^T & y_i' X_i^T \\ X_i^T & 0 & -x_i' X_i^T \\ \vdots & \vdots & \vdots \quad \vdots \end{bmatrix}_{2n\times 9}}_{A} \underbrace{\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}_{9\times 1}}_{H} = 0$$

s.t. $\|h\|=1$ to avoid lazy dummy solution

$AH = 0 \Rightarrow H = A$'s smallest eigen vect

Note that we only have 2-DoF, for a point's corresponding A.

directly from cross product, we have.

$$\underbrace{\begin{bmatrix} 0 & X_i^T & y_i X_i^T \\ X^T & 0 & -x_i X_i^T \\ -yX^T & xX^T & 0 \end{bmatrix}}_{\text{Rank } 2.} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0,$$

but the result is 0, so they are at most 2-linear independent

(row1 $\cdot h_1$ + row2 $\cdot h_2$ + row3 $\cdot h_3 = 0$)

H has 8-DoF, so at least we need 4 points.

Robust feature-based alignment

1. extract feature
2. compute putative matches (corresponding feature pair)
3. Loop:

$$\left.\begin{array}{l} \text{Hypothesis transformation } T \\ \text{Verify} \end{array}\right] \text{RANSAC}$$

Feature Description

Naive Method

Only compare pixel intensity

$\sum (u_i - v_i)^2 \longrightarrow$ normalized $\rho(u,v) = \frac{(u-\bar{u})^T}{\|u-\bar{u}\|} \cdot \frac{(v-\bar{v})}{\|v-\bar{v}\|}$

Invariant to affine transformation

However, feature detectors may not work that good, small deformation could make the problem unsolvable.

↳ Use histogram of gradient. (SIFT, see Lect 7)

Pros:

1. Gradient is not sensitive to illumination changes
2. Histogram can suffer some small transformations

Cons:

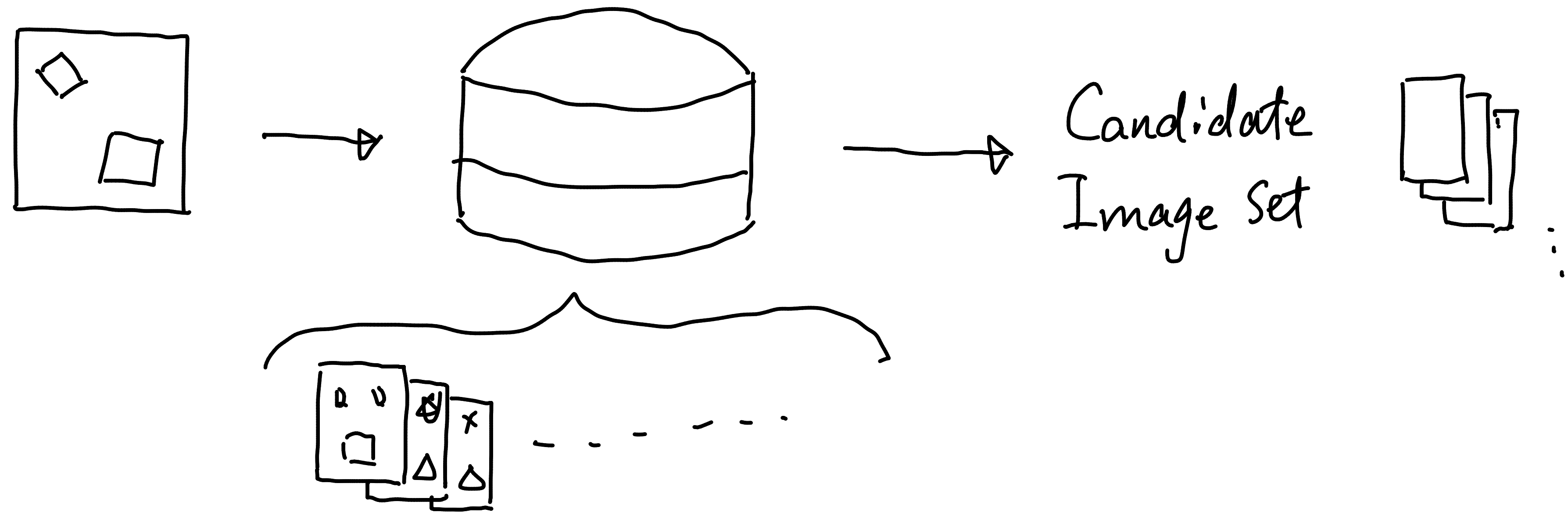
Still sensitive to spacial transformation.

Now, finding putative feature matches can rely mostly on description.

Problem:

1. Ambiguity. If there are many similar features.
   ↳ Reject not unique feature.

Scalability : Find match in a large dataset



Method 1.

The data base stores features, and which model images have them. So, for test image, we extract features, and find out model images which share at least one same feature as test image.

Vocabulary Tree

Measure each feature pair's distance.

Find a representative feature of a group of feature

(Kmean) The final descriptor. (Community Detection)

Every time we get a new test image, find the representative feature and find its correspond images

Hieritical K-mean

Keep run 3-k-mean for sub-region

↳ build a vocabulary tree