CS-565 Computer Vision

Nazar Khan

Department of Computer Science
University of the Punjab

11. Transformations II: Estimation and Warping

Estimation of Affine Transform

- We are given N corresponding points

$$\begin{gathered} \mathbf{x}_1 \Longleftrightarrow \mathbf{x}'_1 \\ \mathbf{x}_2 \Longleftrightarrow \mathbf{x}'_2 \\ \vdots \\ \mathbf{x}_N \Longleftrightarrow \mathbf{x}'_N \end{gathered}$$

  where $\mathbf{x}'_i = \mathbf{T}\mathbf{x}_i$ represents an affinely transformed point pair.

- Goal is to find the 6 parameters

$$\begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix}$$

  of the affine transformation $\mathbf{T}$ that maps the $\mathbf{x}_i$s to $\mathbf{x}'_i$s.

Estimation of Affine Transform

- By writing the transformation parameters in vector form, the ith correspondence $\mathbf{x}'_i = \mathbf{T}\mathbf{x}_i$ can be written as

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ e \\ c \\ d \\ f \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix}$$

Estimation of Affine Transform

- All N correspondences can be written as

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ & & & \vdots & & \\ x_N & y_N & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_N & y_N & 1 \end{bmatrix}}_{2N \times 6} \underbrace{\begin{bmatrix} a \\ b \\ e \\ c \\ d \\ f \end{bmatrix}}_{6 \times 1} = \underbrace{\begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ x'_N \\ y'_N \end{bmatrix}}_{2N \times 1}$$

  which can be seen as a linear system $\mathbf{A}\mathbf{v} = \mathbf{b}$.
- Can be solved via pseudoinverse

$$\mathbf{A}\mathbf{v} = \mathbf{b} \implies \mathbf{A}^T\mathbf{A}\mathbf{v} = \mathbf{A}^T\mathbf{b} \implies \mathbf{v} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \mathbf{A}^\dagger\mathbf{b}$$

  where $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the $6 \times 2N$ matrix called the *pseudoinverse* of $\mathbf{A}$.

Estimation of Affine Transform
Algorithm

Input: N point correspondences $\mathbf{x}_i \Longleftrightarrow \mathbf{x}'_i$

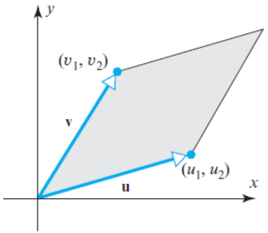
1. Fill in the $2N \times 6$ matrix $\mathbf{A}$ using the $\mathbf{x}_i$.
2. Fill in the $2N \times 1$ vector $\mathbf{b}$ using the $\mathbf{x}'_i$.
3. Compute $6 \times 2N$ pseudo-inverse $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.
4. Compute optimal affine transformation parameters as $\mathbf{v}^* = \mathbf{A}^\dagger\mathbf{b}$.

Detour – Cross Product

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}}_{[\mathbf{u}]_\times} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

- Only defined for 3-dimensional space.
- Matrix $[\mathbf{u}]_\times$ has two linearly independent rows.
  - *Proof*: u_1 row1 $+ u_2$ row2 $+ u_3$ row3 $= \mathbf{0}^T \implies$ any row can be written as a linear combination of the other two rows.
- $\mathbf{u} \times \mathbf{v}$ is another 3-dimensional vector orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.
- $\|\mathbf{u} \times \mathbf{v}\|$ represents the area of the parallelogram formed by $\mathbf{u}$ and $\mathbf{v}$.

Detour – Cross Product



- If **u** and **v** point in the same direction, then no parallelogram will be formed.
- Therefore $\|\mathbf{u} \times \mathbf{v}\|$ will be 0.
- The only vector with norm 0 is the $\mathbf{0}$ vector.
- Therefore, $\mathbf{u} \times \mathbf{v} = \mathbf{0}$ when **u** and **v** point in the same direction.

Estimation of Projective Transform

- We are given N corresponding points

$$\begin{aligned} \mathbf{x}_1 &\Longleftrightarrow \mathbf{x}'_1 \\ \mathbf{x}_2 &\Longleftrightarrow \mathbf{x}'_2 \\ &\vdots \\ \mathbf{x}_N &\Longleftrightarrow \mathbf{x}'_N \end{aligned}$$

  where $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$ represents a projectively transformed point pair.
- Goal is to find the 8 parameters $h_1, h_2 \ldots, h_8$ of the projective transformation $\mathbf{H}$ that maps the $\mathbf{x}$ points to the $\mathbf{x}'$ points.
- Parameter h_9 can be fixed to be 1.
- The ith correspondence can be written as $\mathbf{x}'_i \equiv \mathbf{H}\mathbf{x}_i$ in projective space[1].

[1] Notice that $\mathbf{x}'_i$ can be a scaled version of $\mathbf{H}\mathbf{x}_i$.

Estimation of Projective Transform

- This implies that the 3-dimensional vectors $\mathbf{x}'_i$ and $\mathbf{H}\mathbf{x}_i$ point in the same direction.
- Their cross-product will be the zero vector.

$$\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$$

$$\begin{bmatrix} x'_i \\ y'_i \\ w'_i \end{bmatrix} \times \begin{bmatrix} \mathbf{h}^{1\,T} \\ \mathbf{h}^{2\,T} \\ \mathbf{h}^{3\,T} \end{bmatrix} \mathbf{x}_i = \mathbf{0}$$

where $\mathbf{h}^{jT}$ is the j-th row of $\mathbf{H}$.

- Cross-product can be performed as

$$\begin{bmatrix} 0 & -w'_i & y'_i \\ w'_i & 0 & -x'_i \\ -y'_i & x'_i & 0 \end{bmatrix} \begin{bmatrix} \mathbf{h}^{1\,T}\mathbf{x}_i \\ \mathbf{h}^{2\,T}\mathbf{x}_i \\ \mathbf{h}^{3\,T}\mathbf{x}_i \end{bmatrix} = \mathbf{0}$$

Estimation of Projective Transform

- After matrix-vector multiplication

$$\begin{bmatrix} y_i'\mathbf{h}^{3T}\mathbf{x}_i - w_i'\mathbf{h}^{2T}\mathbf{x}_i \\ w_i'\mathbf{h}^{1T}\mathbf{x}_i - x_i'\mathbf{h}^{3T}\mathbf{x}_i \\ x_i'\mathbf{h}^{2T}\mathbf{x}_i - y_i'\mathbf{h}^{1T}\mathbf{x}_i \end{bmatrix} = \begin{bmatrix} y_i'\mathbf{x}_i^T\mathbf{h}^3 - w_i'\mathbf{x}_i^T\mathbf{h}^2 \\ w_i'\mathbf{x}_i^T\mathbf{h}^1 - x_i'\mathbf{x}_i^T\mathbf{h}^3 \\ x_i'\mathbf{x}_i^T\mathbf{h}^2 - y_i'\mathbf{x}_i^T\mathbf{h}^1 \end{bmatrix} = \mathbf{0}$$

- After separating the unknowns

$$\begin{bmatrix} \mathbf{0}^T & -w_i'\mathbf{x}_i^T & y_i'\mathbf{x}_i^T \\ w_i'\mathbf{x}_i^T & \mathbf{0}^T & -x_i'\mathbf{x}_i^T \\ -y_i'\mathbf{x}_i^T & x_i'\mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix}_{3\times 9} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix}_{9\times 1} = \mathbf{A}_i\mathbf{h} = \mathbf{0}$$

- Matrix $\mathbf{A}_i$ has only 2 linearly independent rows.
- So one row can be discarded. Let's denote the resulting 2×9 matrix by $\mathbf{A}_i$ as well.

Estimation of Projective Transform

- So one correspondence $\mathsf{x}_i \Longleftrightarrow \mathsf{x}'_i$ yields 2 equations.
- Since 8 unknowns require atleast 8 equations, we will need $N \geq 4$ corresponding point pairs.

> The points $\mathsf{x}_1, \ldots, \mathsf{x}_N$ must be non-collinear. Similarly, $\mathsf{x}'_1, \ldots, \mathsf{x}'_N$ must also be non-collinear.

Estimation of Projective Transform

- This will yield the homogenous system $\mathbf{Ah} = \mathbf{0}$ where size of $\mathbf{A}$ is $2N \times 9$.
- It can be shown that $\text{rank}(\mathbf{A}) = 8$ and $\dim(\mathbf{A}) = 9$.
- So nullity of $\mathbf{A}$ is 1 and therefore $\mathbf{h}$ can be found as the null space of $\mathbf{A}$.
- However, when measurements contain noise (which is always the case with pixel locations) or $N > 4$, then no $\mathbf{h}$ will exist that satisfies $\mathbf{Ah} = \mathbf{0}$ exactly.
- In such cases, the best one can do is to find an $\mathbf{h}$ that makes $\mathbf{Ah}$ as close to $\mathbf{0}$ as possible. This can be achieved via

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \|\mathbf{Ah}\|^2 \text{ s.t. } \|\mathbf{h}\|^2 = 1$$

Take-home Quiz 3: Show that $\mathbf{h}^*$ must be the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue.

Estimation of Projective Transform

- This can be done via singular value decomposition.

$$[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \text{svd}(\mathbf{A})$$

and $\mathbf{h}$ is the last column of the matrix $\mathbf{V}$.

Estimation of Projective Transform
Algorithm

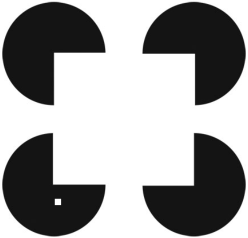
Input: N point correspondences $\mathbf{x}_i \Longleftrightarrow \mathbf{x}'_i$

1. Fill in the $2N \times 9$ matrix $\mathbf{A}$ using the $\mathbf{x}_i$ and $\mathbf{x}'_i$.
2. Compute $[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \text{svd}(\mathbf{A})$.
3. Optimal projective transformation parameters $\mathbf{h}^*$ are the last column of matrix $\mathbf{V}$.

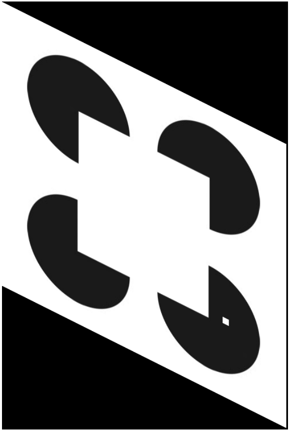
This algorithm is known as the *Direct Linear Transform (DLT)*.[2]

[2] For some practical tips, please refer to slides 14 – 17 from http://www.ele.puc-rio.br/~visao/Homographies.pdf

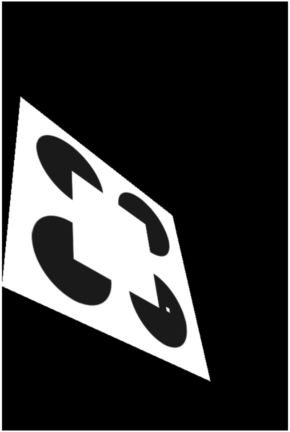
Image Warping



Original



Affine



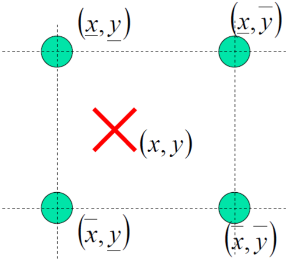
Projective

Image Warping

- Inputs: Image I and transformation matrix $\mathbf{H}$.
- Output: Transformed image $I' = \mathbf{H}I$.
- Obvious approach:
  - For each pixel $\mathbf{x}$ in image I
  - Find transformed point $\mathbf{x}' = \mathbf{H}\mathbf{x}$
  - Divide by 3rd coordinate and move to Cartesian space
  - Copy the pixel color as $I'(\mathbf{x}') = I(\mathbf{x})$.
- Problem: Can leave holes in I'. Why?
- Solution:
  - For each pixel $\mathbf{x}'$ in image I'
  - Find transformed point $\mathbf{x} = \mathbf{H}^{-1}\mathbf{x}'$
  - Divide by 3rd coordinate and move to Cartesian space
  - Copy the pixel color as $I'(\mathbf{x}') = I(\mathbf{x})$.
- Problem: Transformed point $\mathbf{x}$ is not necessarily integer valued.

Image Warping
Bilinear Interpolation

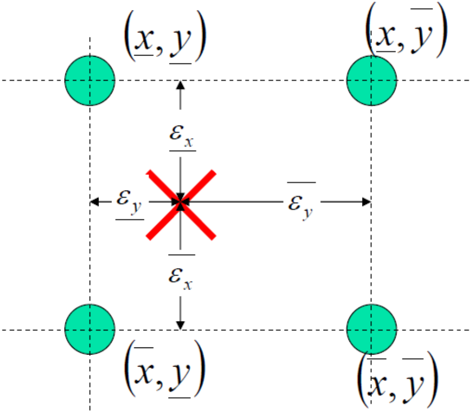
Find 4 nearest pixel locations around (x, y)



where

$$\underline{x} = \lfloor x \rfloor$$
$$\underline{y} = \lfloor y \rfloor$$
$$\bar{x} = \lfloor x \rfloor + 1$$
$$\bar{y} = \lfloor y \rfloor + 1$$

Image Warping
Bilinear Interpolation



$$I(x,y) = \bar{\epsilon}_x\bar{\epsilon}_y I(\underline{x},\underline{y}) + \underline{\epsilon}_x\bar{\epsilon}_y I(\bar{x},\underline{y}) + \bar{\epsilon}_x\underline{\epsilon}_y I(\underline{x},\bar{y}) + \underline{\epsilon}_x\underline{\epsilon}_y I(\bar{x},\bar{y})$$