Numerical Example

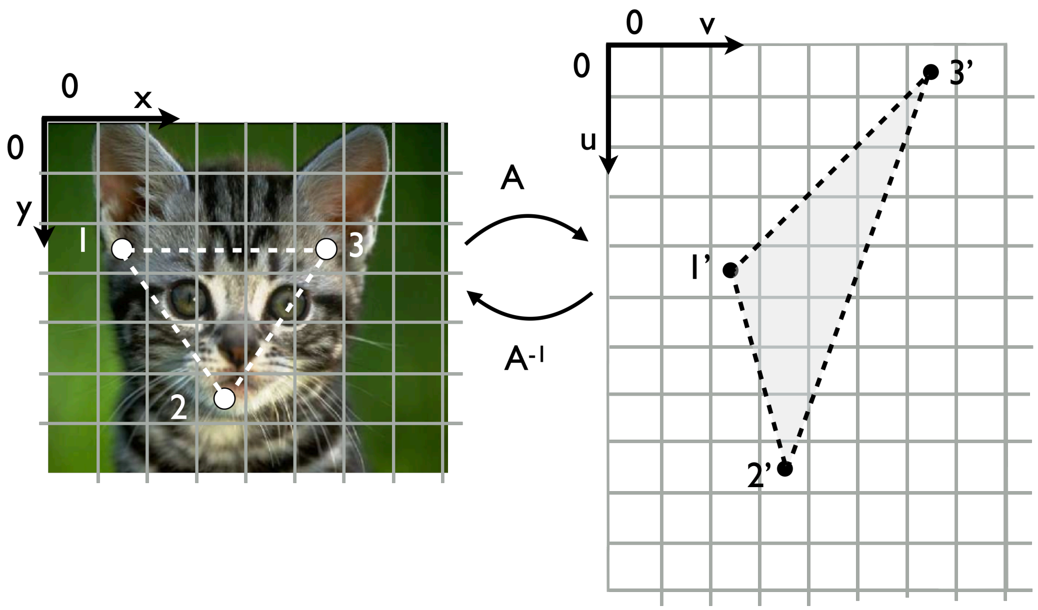

Figure 1: A pair of triangles related by the affine transformation A.

1 Estimating the matrix of the affine transformation

There are two coordinate systems, i.e., $xy-$coordinate system and the $uv-$coordinate system. To find the affine transformation relating the pair of triangles, we need to solve the following system of equations:

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}}_{\mathbf{b}} \tag{12}$$

or, in short:

$$M\mathbf{a} = \mathbf{b}. \tag{13}$$

To solve Equation 13 for $\mathbf{a}$, we must invert the matrix on the left-hand side, i.e.:

$$\mathbf{a} = M^{-1}\mathbf{b}. \tag{14}$$

1.1 Algorithm

```
function A = ESTIMATEAFFINE(x, x')                ▷ 3 pairs of corresponding vertices
    Create matrices M and b using the 3 pairs of vertices
    a ← M^{-1} * b                                            ▷ Affine coefficients
    A ← reshape(a, 3 × 3)                                ▷ Create the affine matrix
    return A
end function
```

1.2 Measurements

The vertices of the triangle in the $xy-$coordinate system are given by $\mathbf{p}_1 = (1,2)^\top$, $\mathbf{p}_2 = (3,5)^\top$, and $\mathbf{p}_3 = (5,2)^\top$, which can be arranged as columns of a matrix representing the triangle, i.e.:

$$T_{xy} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3] = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 5 & 2 \end{bmatrix}. \tag{15}$$

Similarly, the vertices of the triangle in the $uv-$coordinate system are $\mathbf{p}'_1 = (2,4)^\top$, $\mathbf{p}'_2 = (3,8)^\top$, and $\mathbf{p}'_3 = (6,0)^\top$, which can be arranged as columns of a matrix representing the triangle, i.e.:

$$T_{uv} = [\mathbf{p}'_1 \quad \mathbf{p}'_2 \quad \mathbf{p}'_3] = \begin{bmatrix} 2 & 3 & 6 \\ 4 & 8 & 0 \end{bmatrix}. \tag{16}$$

1.3 Creating the system of equations and solving it

Substitute the measurements to form matrix M and matrix $\mathbf{b}$:

$$M = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 3 & 5 & 1 & 0 & 0 & 0 \\ 5 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 3 & 5 & 1 \\ 0 & 0 & 0 & 5 & 2 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 2 \\ 3 \\ 6 \\ 4 \\ 8 \\ 0 \end{bmatrix} \tag{17}$$

Solution:

$$\mathbf{a} = M^{-1}\mathbf{b} = \begin{bmatrix} 1 \\ -\frac{1}{3} \\ \frac{5}{3} \\ -1 \\ 2 \\ 1 \end{bmatrix}. \tag{18}$$

Use the values in $\mathbf{a}$ to fill in the A transformation matrix:

$$A = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{5}{3} \\ -1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

1.4 Testing the affine transformation

Mapping from (x, y) to (u, v):

$$A\tilde{T}_{xy} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{5}{3} \\ -1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 5 \\ 2 & 5 & 2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 6 \\ 4 & 8 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{20}$$

Here, we use tilde to indicate that the measurements are in homogeneous coordinates $\tilde{T}_{xy}$. Similarly, we can (inverse) map from (u, v) to (x, y) using:

$$A^{-1}\tilde{T}_{uv} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{5}{3} \\ -1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 3 & 6 \\ 4 & 8 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 5 & 2 \\ 1 & 1 & 1 \end{bmatrix}. \tag{21}$$

2 Performing inverse mapping to transfer colors enclosed by the two shapes

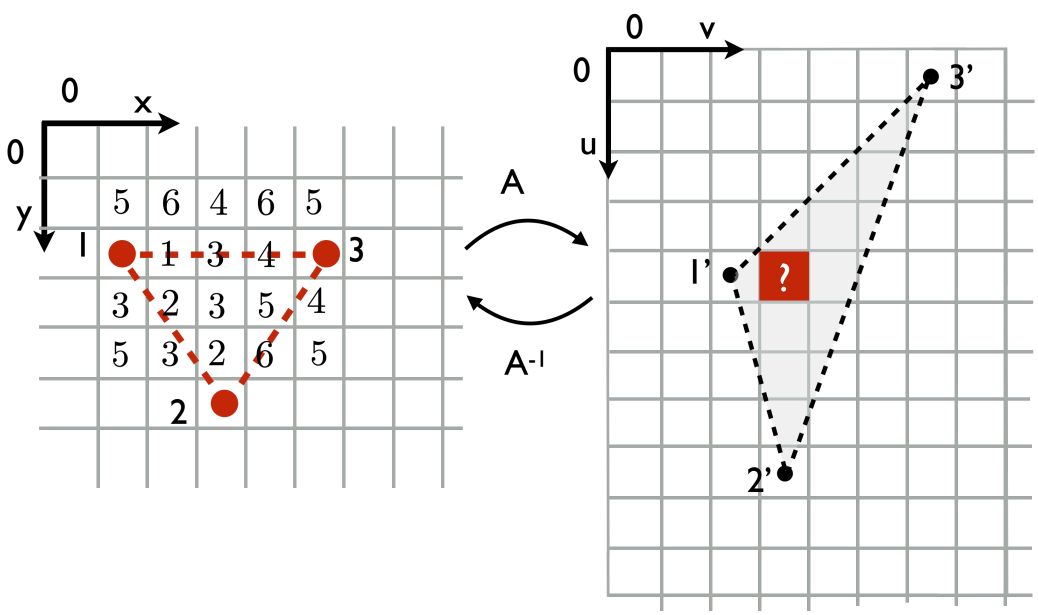

Figure 2: A pair of triangles related by the affine transformation A. Colors are transferred from the source image to the destination image using inverse mapping.

2.1 Algorithm

function $I_{dst} = \text{INVERSEMAPPING}(A, \mathbf{x}, \mathbf{x}', I_{src})$
  $n_{rows}, n_{cols} \leftarrow getDstImageSize(\mathbf{x}')$ ▷ Obtain size of destination image
  $I_{dst} \leftarrow zeros(n_{rows}, n_{cols})$ ▷ Initialize destination image
  for all $\mathbf{x}'$ **do** ▷ Loop over all pixels is dst image
    $\mathbf{x} \leftarrow A^{-1} * \mathbf{x}'$ ▷ Obtain pixel coords from dst image to src image
    $I_{dst}(\mathbf{x}') \leftarrow I_{src}(\mathbf{x})$ ▷ Copy pixel color from src image to dst
  end for
  return I_{dst}
end function

function $n_{rows}, n_{cols} = \text{GETDSTIMAGESIZE}(\mathbf{x}')$
  $x_{min} \leftarrow min(x_i')$ ▷ Max. and Min. of polygon boundaries
  $x_{max} \leftarrow max(x_i')$
  $y_{min} \leftarrow min(y_i')$
  $y_{max} \leftarrow max(y_i')$
  $n_{rows} \leftarrow y_{max} - y_{min}$ ▷ Num of rows (i.e., height)
  $n_{cols} \leftarrow x_{max} - x_{min}$ ▷ Num of cols (i.e., width)
  return n_{rows}, n_{cols}
end function

2.2 Performing inverse mapping on a single pixel

Considering the triangles shown in Figure 2, the question we need to answer is: What color should go into pixel $(3,4)^\mathsf{T}$ in the $uv-$coordinate system? Answer: The color is at the corresponding location of that pixel in the xy-coordinates when we map back to the original image, i.e., we use the inverse transformation to take us to that location:

$$A^{-1}\begin{bmatrix}3\\4\\1\end{bmatrix} = \begin{bmatrix}1 & -\frac{1}{3} & \frac{5}{3}\\ -1 & 2 & 1\\ 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}3\\4\\1\end{bmatrix} = \begin{bmatrix}1 & 3 & 5\\ 2 & 5 & 2\\ 1 & 1 & 1\end{bmatrix} = \begin{bmatrix}2.2\\2.6\\1.0\end{bmatrix}. \tag{22}$$

Because the matrix of the source image is indexed by integers, we will round the coordinates to $(2,3)^\mathsf{T}$. In the example shown in Figure 2, the color value is 2.

3 Main algorithm for affine warping

procedure AFFINEWARP( )
Require:
  I_{src} ▷ Source image
  $\mathbf{x} = \{(x_i, y_i, 1)\}_{i=1}^{3}$ ▷ Vertices from source triangle
  $\mathbf{x}' = \{(x_i', y_i', 1)\}_{i=1}^{3}$ ▷ Vertices from destination triangle

  $A \leftarrow estimateAffine(\mathbf{x}, \mathbf{x}')$ ▷ Estimate transformation between two triangles
  $I_{dst} \leftarrow inverseMapping(A, \mathbf{x}, \mathbf{x}', I_{src})$ ▷ Warp pixels from src to dst image
  $displayImage(I_{dst})$
end procedure