# Digital Image Processing

Abhijit Amrendra Kumar

August 2023

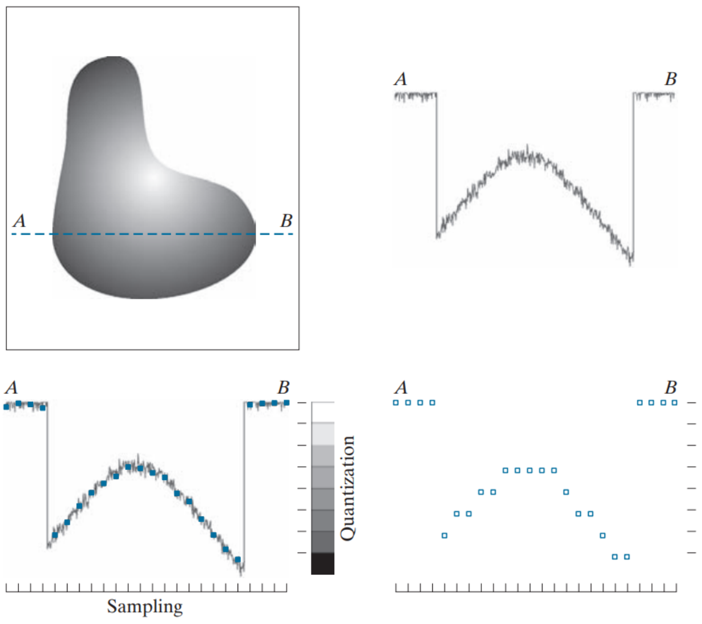
# Chapter 1

# Image Warping and Alignment

## 1.1 Basics

A digital image, which is a version of the visual stimulus sampled at discrete locations with discretized values, can be regarded as a function $I = f(x, y)$, where $(x, y)$ are the spatial integer coordinates in typically a rectangular domain $\Omega = [0, W - 1] \times [0, H - 1]$. Each ordered pair $(x, y)$ denote a pixel, which is generally a square. Pixel dimensions relate to the spatial resolution of the light-collecting sensor of a camera.

The actual visual signal is analog, but digital cameras capture a discrete version of it, and also quantize the intensity values. In a typical grayscale image, the intensity values lie from $0 \rightarrow 255$



## 1.2 Alignment

Consider images $I_1, I_2$ of a scene acquired through different viewpoints. $I_1$ and $I_2$ are said to be aligned iff for every $(x, y)$ in the domain $\Omega$, the pixels at $(x, y)$ in $I_1$ and $I_2$ are in physical correspondence.

If not, the images are said to be misalgined with respect to each other, or we say there is relative motion between the images. Image alignment is the process of correcting for the relative motion between $I_1$ and $I_2$.

## 1.3 Motion Models

Let us denote the coordinates in $I_1$ as $(x_1, y_1)$, and those in $I_2$ as $(x_2, y_2)$.

Let's first consider **translation**.

$$\forall (x_1, y_1) \in \Omega, \exists t_x, t_y, x_2 = x_1 + t_x, y_2 = y_1 + t_y \tag{1.1}$$
$$\implies \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{1.2}$$

Next, let's consider rotation about a point $(x_c, y_c)$ anti-clockwise through angle $\theta$

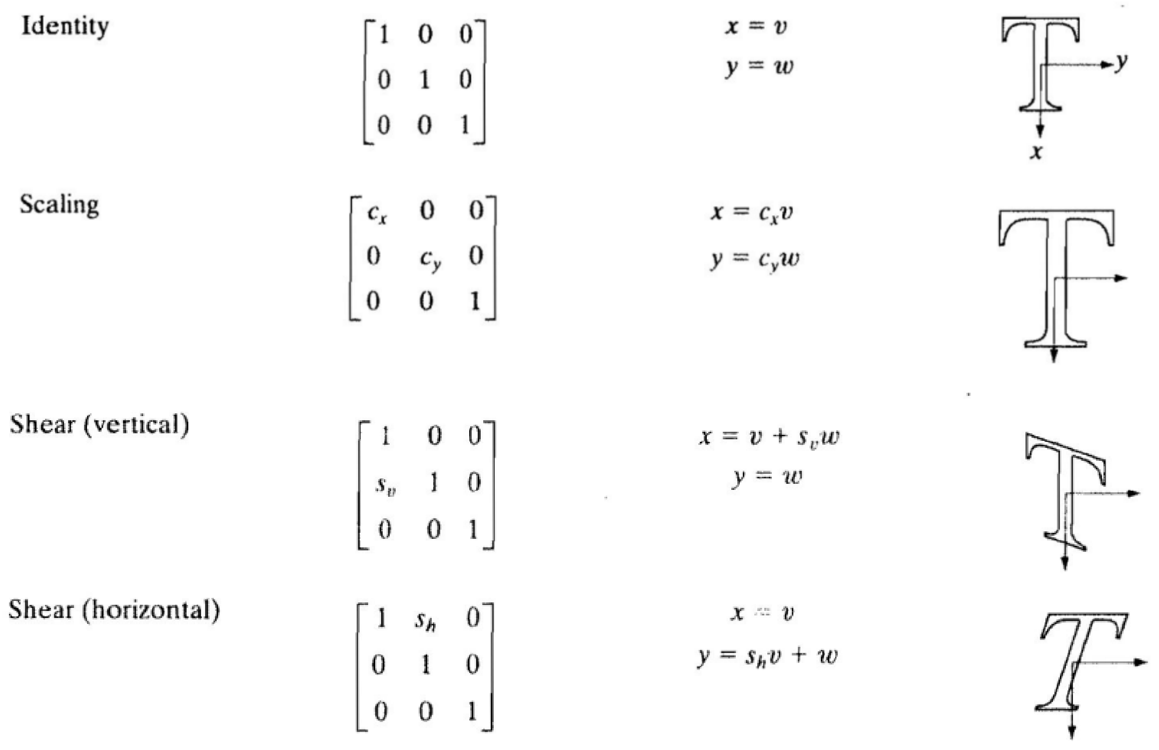
$$x_2 = (x_1 - x_c)\cos\theta - (y_1 - y_c)\sin\theta + x_c \tag{1.3}$$
$$y_2 = (x_1 - x_c)\sin\theta + (y_1 - y_c)\cos\theta + y_c \tag{1.4}$$
$$\implies \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_c \\ \sin\theta & \cos\theta & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 - x_c \\ y_1 - y_c \\ 1 \end{bmatrix} \tag{1.5}$$

Combining the two motions, we obtain

$$x_2 = (x_1 - x_c)\cos\theta - (y_1 - y_c)\sin\theta + t_x \tag{1.6}$$
$$y_2 = (x_1 - x_c)\sin\theta + (y_1 - y_c)\cos\theta + t_y \tag{1.7}$$
$$\implies \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 - x_c \\ y_1 - y_c \\ 1 \end{bmatrix} \tag{1.8}$$

If the $2 \times 2$ matrix $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ is rank-deficient, we term the transformation as an **affine transformation**, since it will transform 2D figures into a line/point.

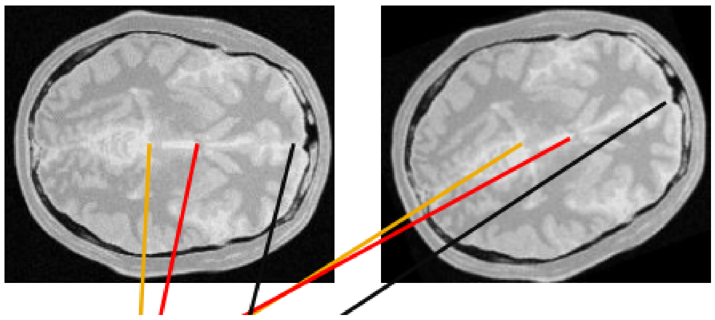
**Note**: Composition of multiple types of motion is given by the multiplication of their corresponding matrices. However, most motion compositions are not commutative.



| | | | |
|---|---|---|---|
| Identity | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v$<br>$y = w$ | |
| Scaling | $\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = c_x v$<br>$y = c_y w$ | |
| Shear (vertical) | $\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v + s_v w$<br>$y = w$ | |
| Shear (horizontal) | $\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v$<br>$y = s_h v + w$ | |

## 1.4 Image Alignment: Control Points

This method involves marking pairs of physically corresponding points on two images, either manually by an expert or automatically using geometric properties. Then using these points, the transformation matrix can be computed, and the whole image can be transformed to align with the original image.



However, this method is always not feasible, because

- it requires manual intervention
- it is error prone
- there are automatic methods like the SIFT technique to finding matching control points.

## 1.5 Image Alignment: Mean Squared Error

The mean squared error between 2 images is given by

$$\text{MSSD} = \frac{1}{N}\sum_{x,y\in\Omega}(I_1(x,y) - I_2(x,y))^2,\ N = \#\text{pixels in field of view}$$

We can modify the equation in the following manner to find the transform matrix $T$

$$T = \text{argmin } \text{MSSD}_T(I_1(v), I_2(Tv)), \quad v = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$