Beeldverwerken Lab1

Hannah Min, Harm Manders

April 18, 2018

1 Instructions for the MATLAB code

To get the answers to the programming questions, run main.m

2 Interpolation

2.1

$F(x) = floor(x + 0.5)$

2.2

$a = F(k + 1) - F(k)$
$b = (k + 1)F(k) - k \cdot F(k + 1)$

2.3

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k & k+1 \end{bmatrix} \begin{bmatrix} F(k+1) \\ F(k) \end{bmatrix}$$

To fit a line through the points (k, F(k)) and (k+1, F(k+1)), we calculate a and b for the expression f(x) = ax + b. Here, a is defined as the difference in y-direction between the two points, divided by the difference in the x-direction, which is 1 in this case. The value for b can be found by filling in the value for a in combination with one of the two given coordinates. This results in the following expression:
$f(x) = (F(k + 1) - F(k))x + (k + 1)F(k) - k \cdot F(k + 1)$
$f(x) = F(k + 1)x - F(k)x + (k + 1)F(k) - k \cdot F(k + 1)$
$f(x) = (k + 1 - x)F(k) + (x - k)F(k + 1)$

2.4

Given $f(x) = (k + 1 - x)F(k) + (x - k)F(k + 1)$ and $\alpha = x - k$, interpolating between the points F(k, l) and F(k+1, l) results in:
$f(x, l) = (1 - \alpha)F(k, l) + \alpha \cdot F(k + 1, l)$

Interpolating between F(k, l+1) and F(k+1, l+1) results in:
$f(x, l+1) = (1-\alpha)F(k, l+1) + \alpha \cdot F(k+1, l+1)$

$f(x, y) = (1-\beta)f(x, l) + \beta \cdot f(x, l+1)$
$f(x, y) = (1-\beta)((1-\alpha)F(k, l) + \alpha \cdot F(k+1, l)) + \beta((1-\alpha)F(k, l+1) + \alpha \cdot F(k+1, l+1))$

$f(x, y) = (1-\alpha)(1-\beta)F(k, l) + \alpha(1-\beta)F(k+1, l) + \beta(1-\alpha)F(k, l+1) + \alpha\beta \cdot F(k+1, l+1)$

3 Rotation

3.1

$R = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}$

3.2

$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

3.3

$\vec{x'} = R \cdot (\vec{x} - \vec{c}) + \vec{c}$

3.4

It stays the same, as can be derived as follows:

$\vec{c'} = R \cdot (\vec{c} - \vec{c}) + \vec{c} = R \cdot \vec{0} + \vec{c} = \vec{0} + \vec{c} = \vec{c}$

4 Affine Transformations

4.1

The transformation matrix will be of the size 2x3 and the coordinate vector will be of size 2x1. The dimensions don't correspond, so it won't be possible to multiply them.

4.2

When using homogeneous coordinates, we add a third dimension and set it to 1. The last row of the matrix should be [0 0 1], because in this way the

multiplication will still result in the generic formula for affine transformation.
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

4.3

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\phi) & -sin(\phi) & c \\ sin(\phi) & cos(\phi) & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

It performs a rotation ϕ and a translation $\langle c, f \rangle^T$

4.4

The corner points of a desired parallelogram are easy to find in the original image. Therefore we suggest using the origin, the corner on the x-axis and the corner on the y-axis as output coordinates. The input coordinates can be assigned by clicking on the corresponding corners in the original image.

4.5

An affine transformation has 6 degrees of freedom: translation in two directions, scaling in two directions, rotating and shearing. To find the values of the 6 parameters, 6 equations are needed. Since each pair of coordinates results in 2 equations, we need three point pairs. Therefore this is the minimum, but more pairs would be inconvenient, because it is hard to select points that correspond perfectly to how the parallelogram is already defined.

5 Re-projecting Images

5.1

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}u + m_{12}v + m_{13} \\ m_{21}u + m_{22}v + m_{23} \\ m_{31}u + m_{32}v + m_{33} \end{bmatrix} = \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix}$$

5.2

$\lambda x = m_{11}u + m_{12}v + m_{13}$
$\lambda y = m_{21}u + m_{22}v + m_{23}$
$\lambda = m_{31}u + m_{32}v + m_{33}$

$(m_{31}u + m_{32}v + m_{33})x = m_{11}u + m_{12}v + m_{13}$
$m_{31}ux + m_{32}vx + m_{33}x = m_{11}u + m_{12}v + m_{13}$
$m_{11}u + m_{12}v + m_{13} - m_{31}ux - m_{32}vx - m_{33}x = 0$

$(m_{31}u + m_{32}v + m_{33})y = m_{21}u + m_{22}v + m_{23}$
$m_{31}uy + m_{32}vy + m_{33}y = m_{21}u + m_{22}v + m_{23}$
$m_{21}u + m_{22}v + m_{23} - m_{31}uy - m_{32}vy - m_{33}y = 0$

Filling in λ results in two equations for each point correspondence.

5.3

There are 8 parameters.

5.4

On top of the 6 degrees of freedom from an affine transformation there are two projection parameters m_{31} and m_{32}. The parameter m_{33} is a scalar and doesn't correspond to a degree of freedom.

5.5

To determine a projection matrix, 4 point pairs are needed in 2D and 8 point pairs in 3D.

5.6

$$\vec{x} = \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix}$$

5.7

$$A\vec{x} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 & -x_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -y_1u_1 & -y_1v_1 & -y_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -x_2v_2 & -x_2 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -y_2u_2 & -y_2v_2 & -y_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -x_3v_3 & -x_3 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -y_3u_3 & -y_3v_3 & -y_3 \\ u_4 & v_4 & 1 & 0 & 0 & 0 & -x_4u_4 & -x_4v_4 & -x_4 \\ 0 & 0 & 0 & u_4 & v_4 & 1 & -y_4u_4 & -y_4v_4 & -y_4 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

5.8

This can be avoided by finding the nullspace.

5.9

As stated in 5.4 m_{33} doesn't have an impact. Therefore the matrix can be normalized through dividing it by m_{33}.

5.10

It is sufficient to find the solution to $A\vec{x} = 0$.

6 Questions and Exercises

7 Estimating a Camera's Projection Matrix

7.1

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$\lambda x = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$
$\lambda y = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$
$\lambda = m_{31}X + m_{32}Y + m_{33}Z + m_{34}$

$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})x = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$
$m_{31}Xx + m_{32}Yx + m_{33}Zx + m_{34}x = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$
$m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}Xx - m_{32}Yx - m_{33}Zx - m_{34}x = 0$

$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})y = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$
$m_{31}Xy + m_{32}Yy + m_{33}Zy + m_{34}y = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$
$m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}Xy - m_{32}Yy - m_{33}Zy - m_{34}y = 0$

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1x_1 & -Y_1x_1 \\ -Z_1x_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1y_1 & -Y_1y_1 \\ -Z_1y_1 & -y_1 \\ \dots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -X_nx_n & -Y_nx_n \\ -Z_nx_n & -x_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_ny_n & -Y_ny_n \\ -Z_ny_n & -y_n \end{bmatrix}$$

7.2

For each point pair, the matrix A will increase by two rows. Therefore it can still represent the resulting equations.

7.3

There won't be an exact solution, so the strategy will be to minimize the error.

7.4

To find the smallest $\|A\vec{m}\|$, we say that $\|\vec{m}\| = 1$. The SVD of A is UDV^T, therefore we want to miminize $\left\|UDV^T\vec{m}\right\|$. Since U is orthogonal, it won't have an influence on the length of the vector and can be left out. This results in $\left\|DV^T\vec{m}\right\|$. V is also orhtogonal, which means that $\left\|V^T\vec{m}\right\| = 1$. We can rewrite the previous to $\|Dy\|$, with $y = V^T\vec{m}$. D is the diagonal matrix and has his values sorted from highest to lowest. Therefore $\|Dy\|$ will be the smallest if $y = [0\ 0\ \ldots\ 0\ 1]^T$.
$\vec{m} = Vy$, so $\vec{m} =$ V$[0\ 0\ \ldots\ 0\ 1]^T$. This means that $\vec{m}$ is the last column of V.