Photogrammetric Computer Vision

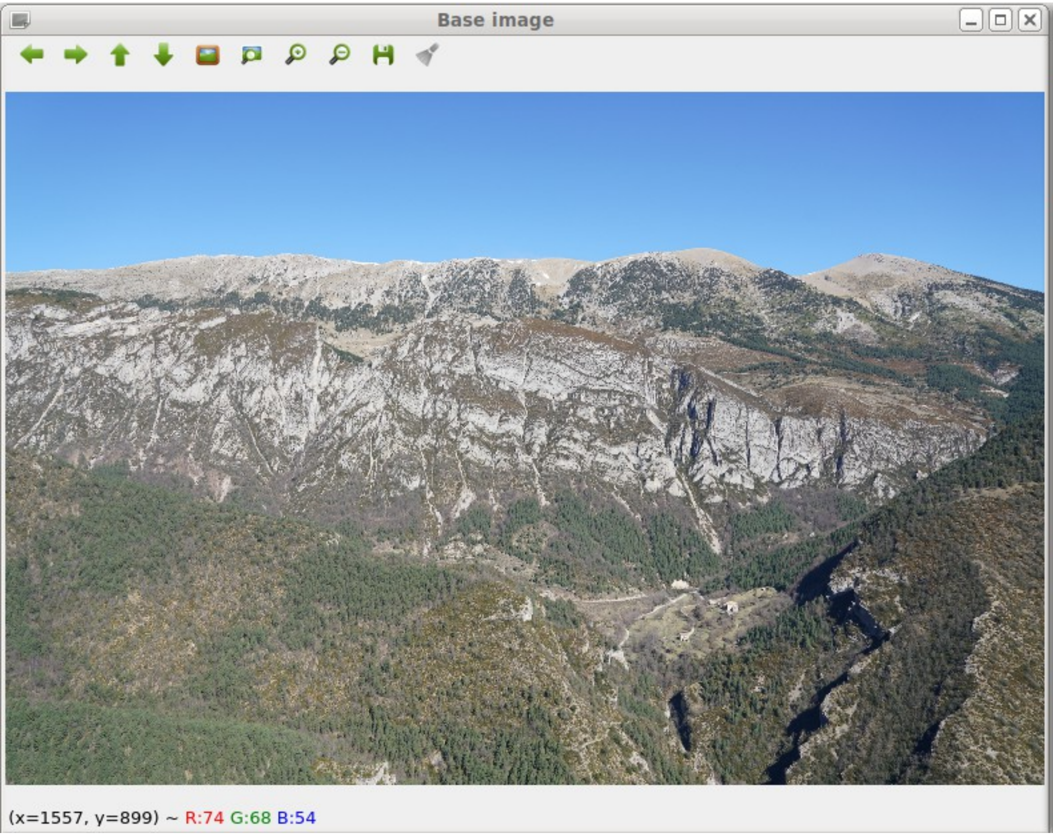
Berlin University of Technology (TUB),
Computer Vision and Remote Sensing Group
Berlin, Germany

2. Exercise

Projective Transformation (Homography)

In this exercise a fundamental *projective transformation* should be realized. The task consists of rectifying images geometrically and stitching them together, so that a *panorama mosaic* is generated.

2. Exercise


Base image
(x=1557, y=899) ~ R:74 G:68 B:54

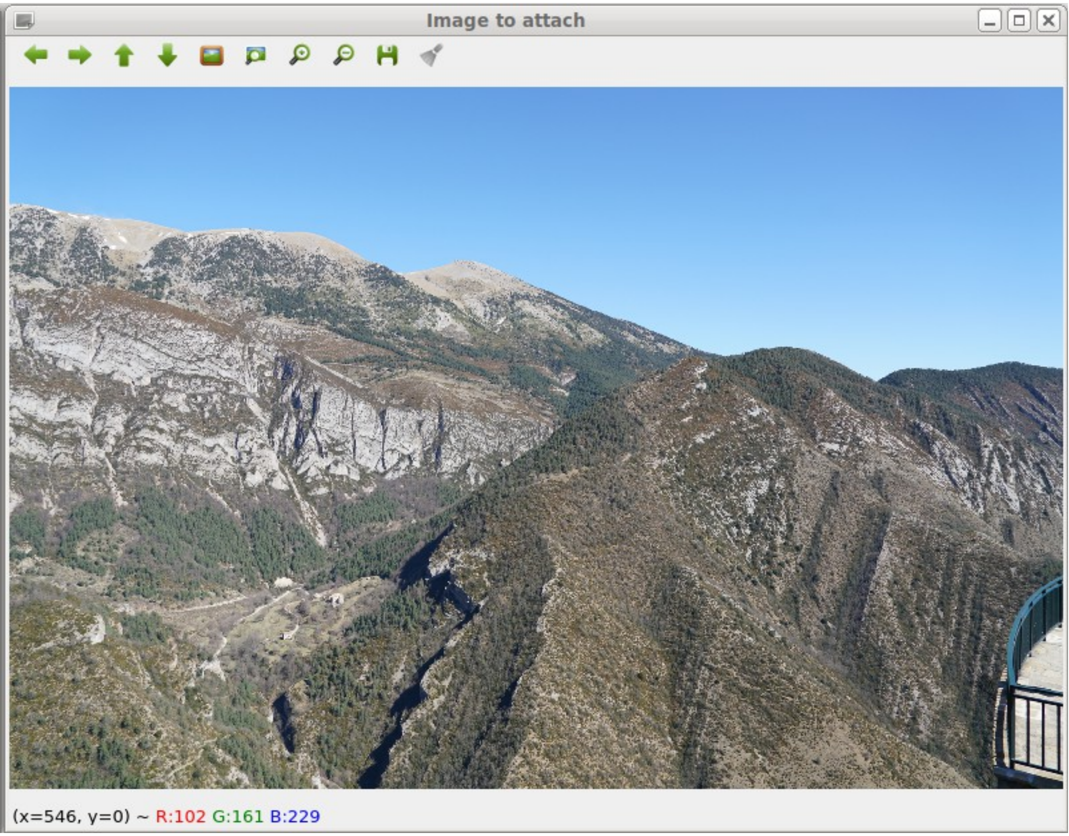
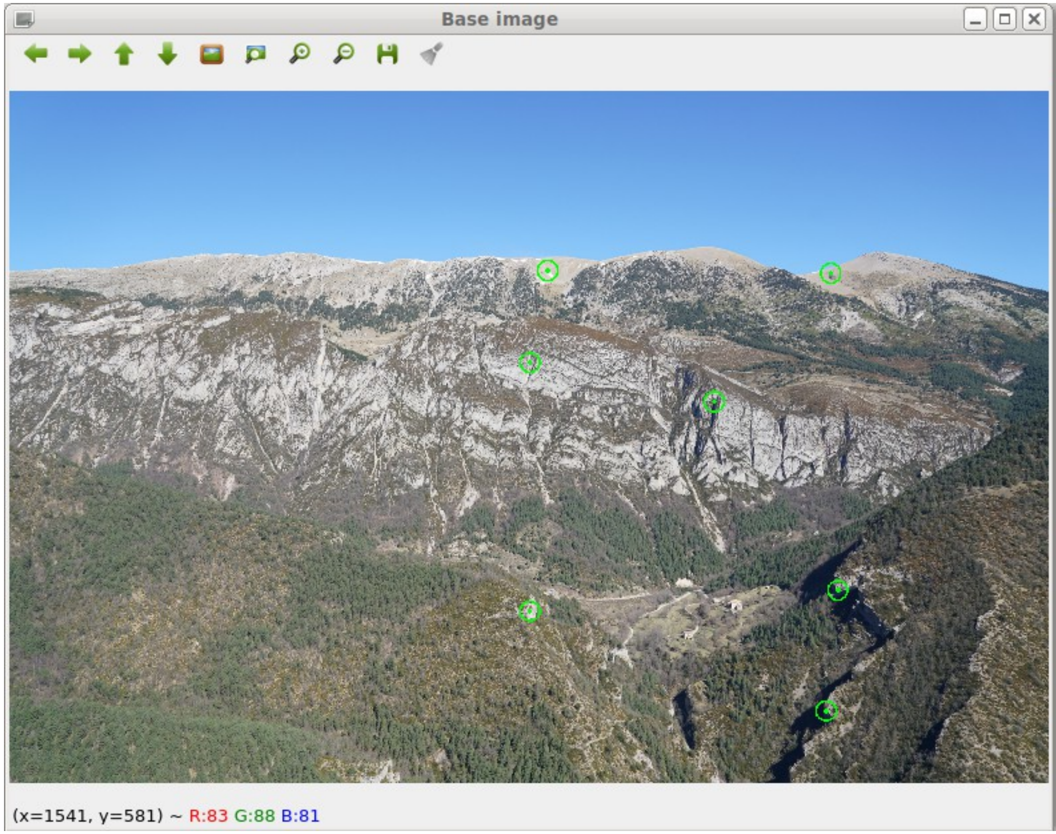


Image to attach
(x=546, y=0) ~ R:102 G:161 B:229

2. Exercise



Base image
(x=1541, y=581) ~ R:83 G:88 B:81

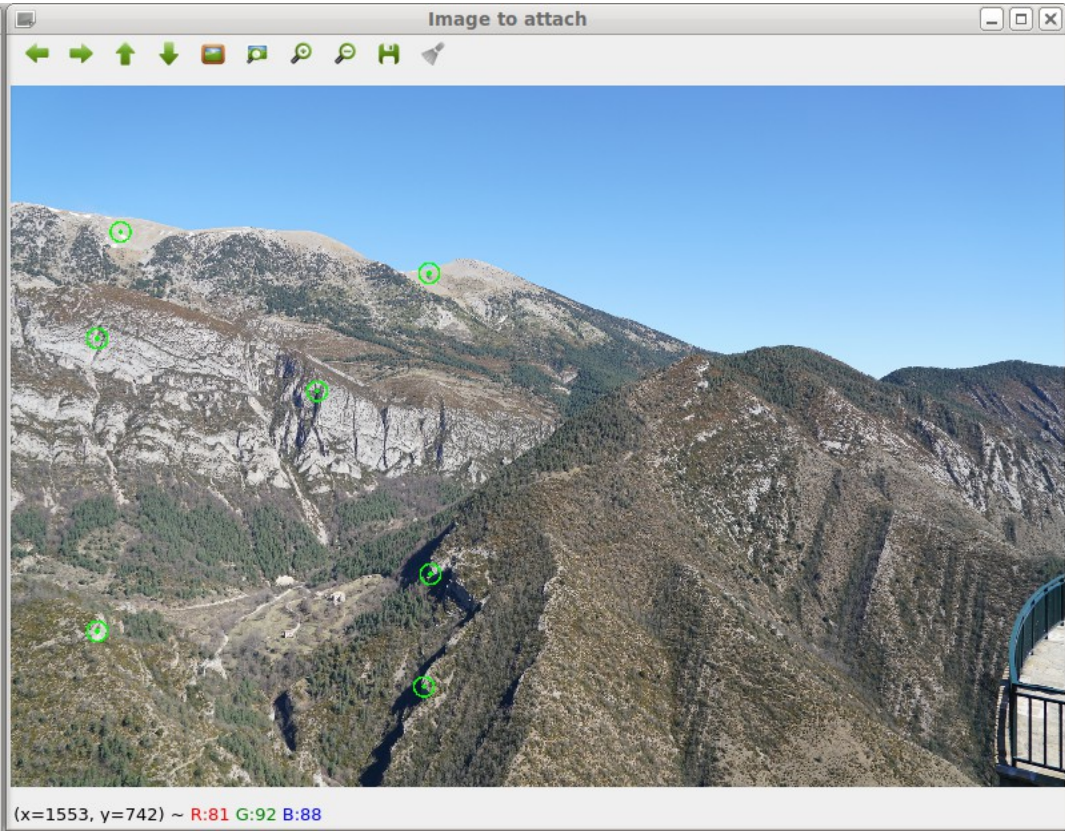
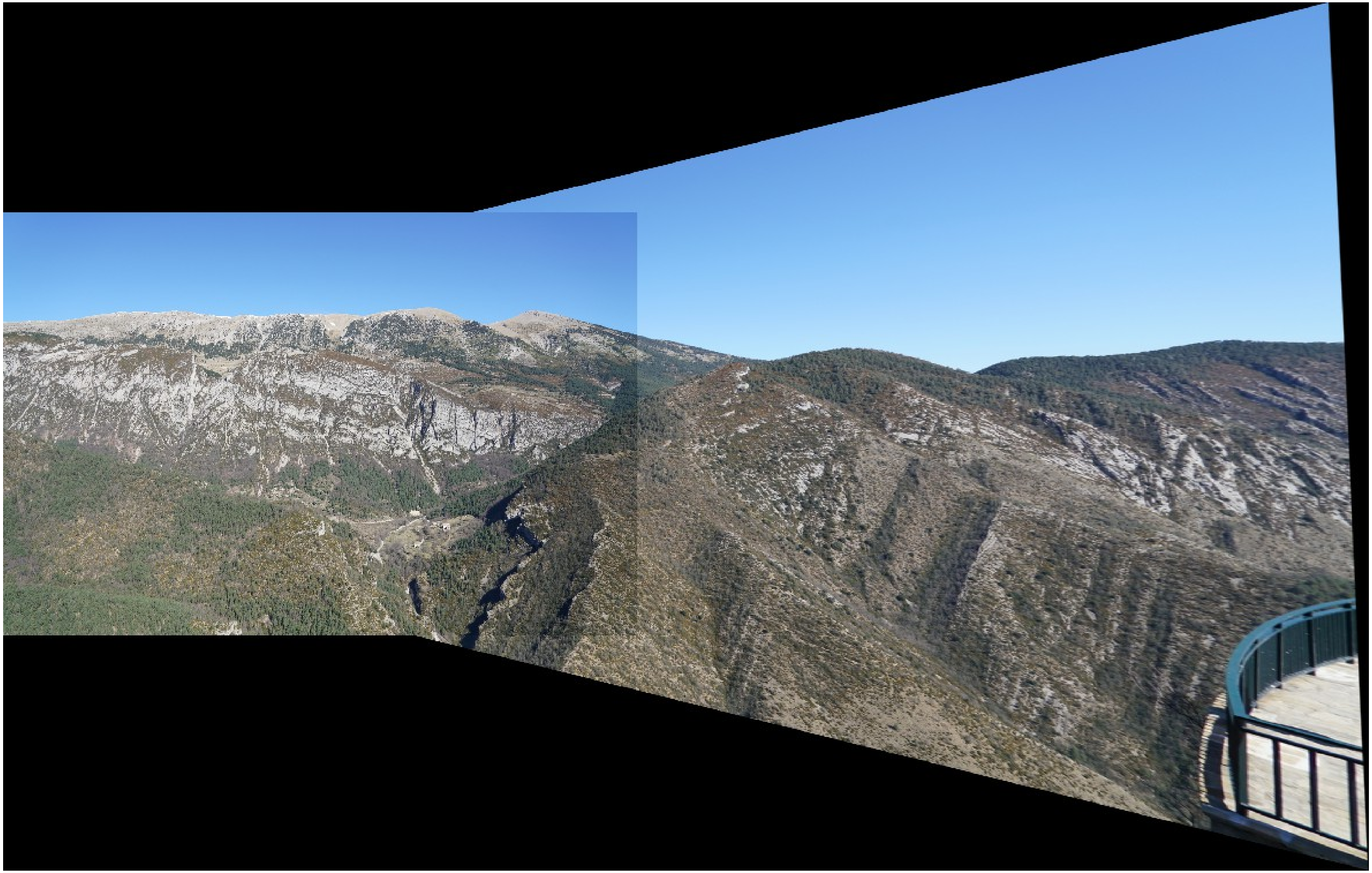


Image to attach
(x=1553, y=742) ~ R:81 G:92 B:88

2. Exercise



2. Exercise

1. Image Acquisition:

Take pictures with a digital camera,
which overlap horizontal at least 30 percent.

Produce three images without disparities
(e.g. turn around the projection centre or choose a planar object).

2. Exercise

1. Image Acquisition:

Take pictures with a digital camera,
which overlap horizontal at least 30 percent.

Produce three images without disparities
(e.g. turn around the projection centre or choose a planar object).

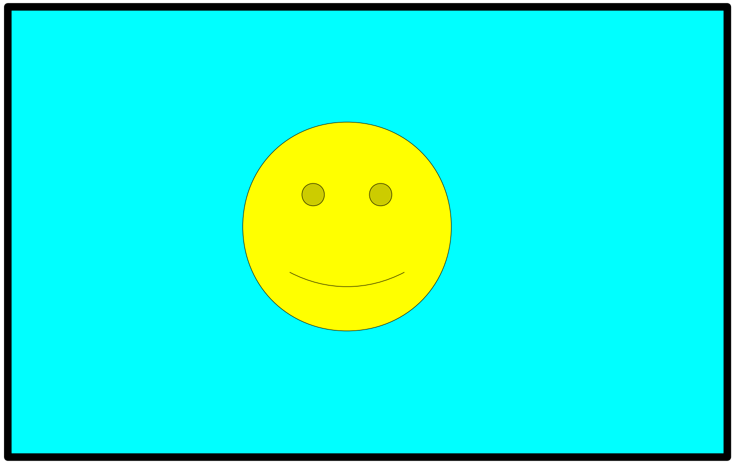
2. Correspondence Analysis:

Transfer the three images into the computer and measure interactively at least four corresponding image points $x \leftrightarrow x'$ between two neighbouring images.

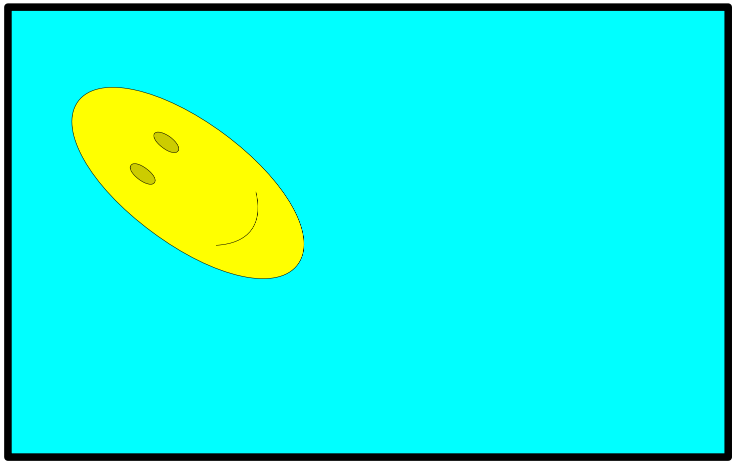
2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using singular value decomposition. Compute the two necessary homography matrices.



First image

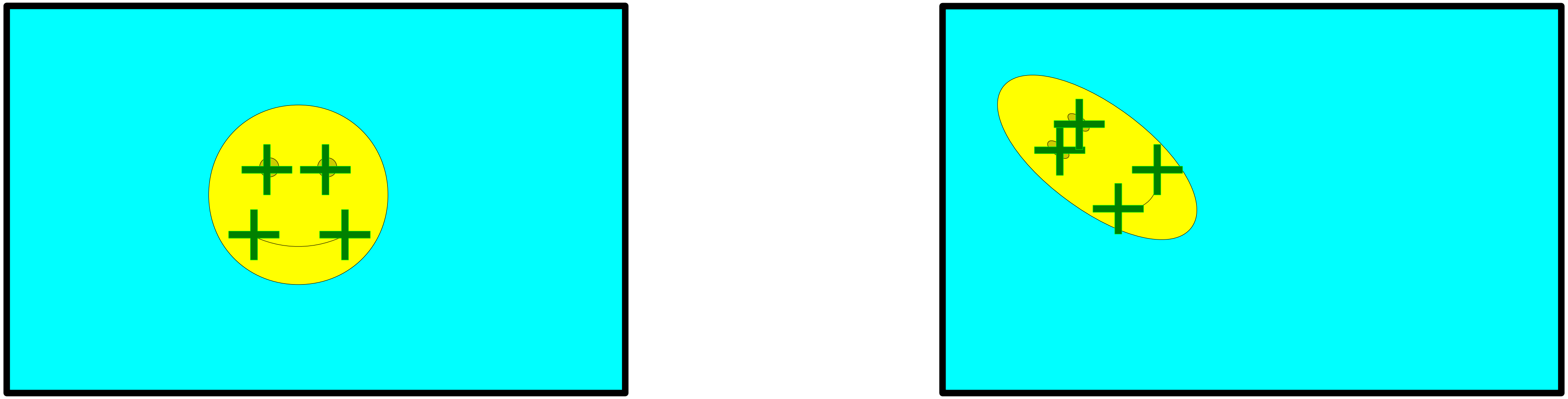


Second image

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using singular value decomposition. Compute the two necessary homography matrices.

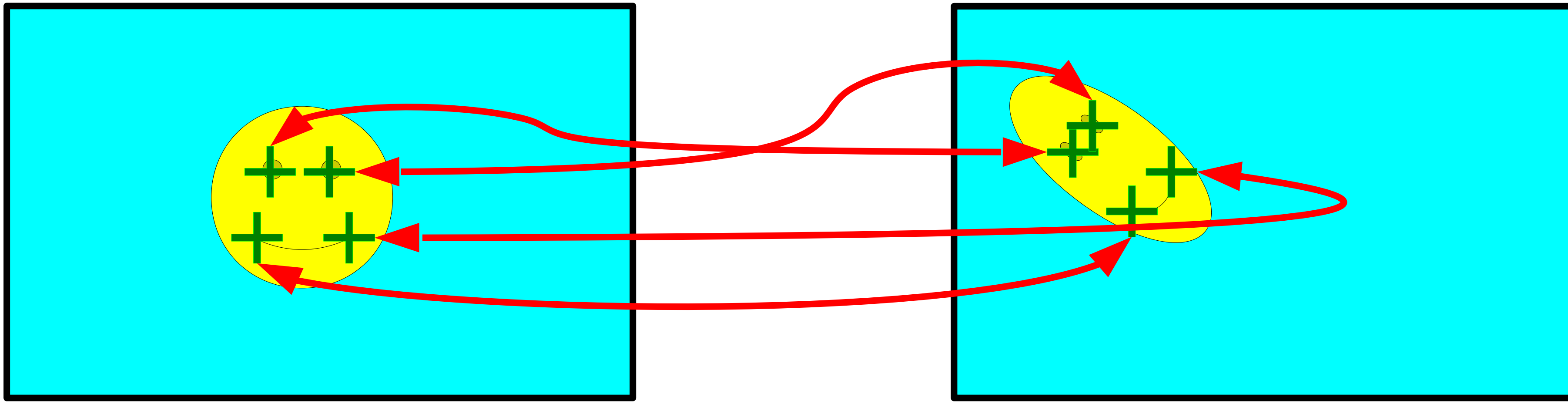


Measure image point pairs

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using singular value decomposition. Compute the two necessary homography matrices.

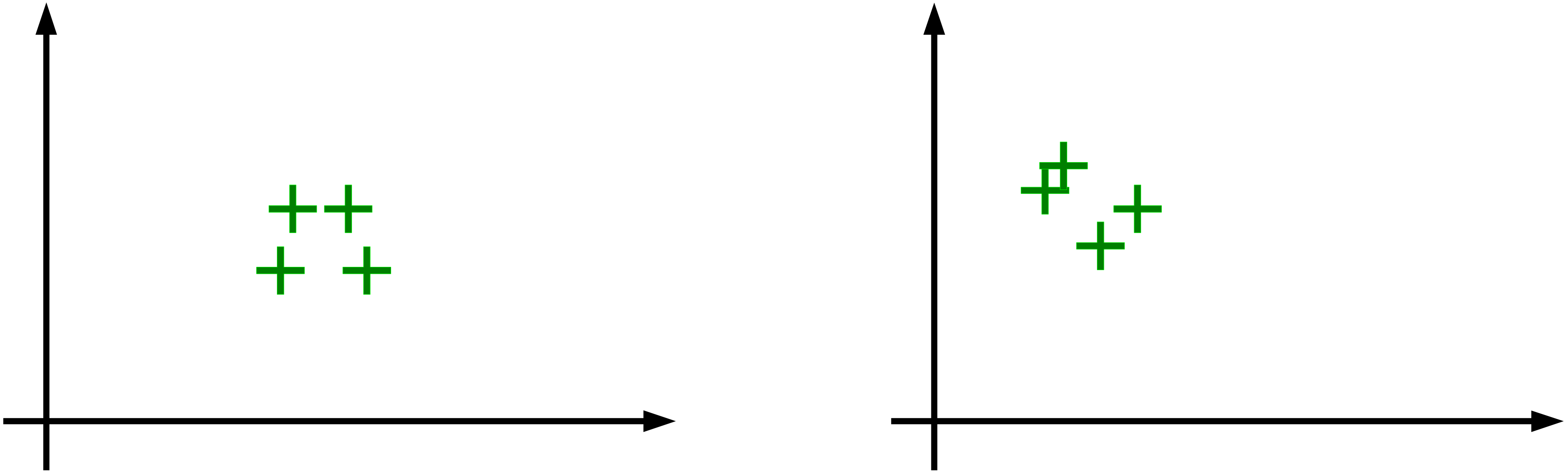


Measure image point pairs

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using singular value decomposition. Compute the two necessary homography matrices.

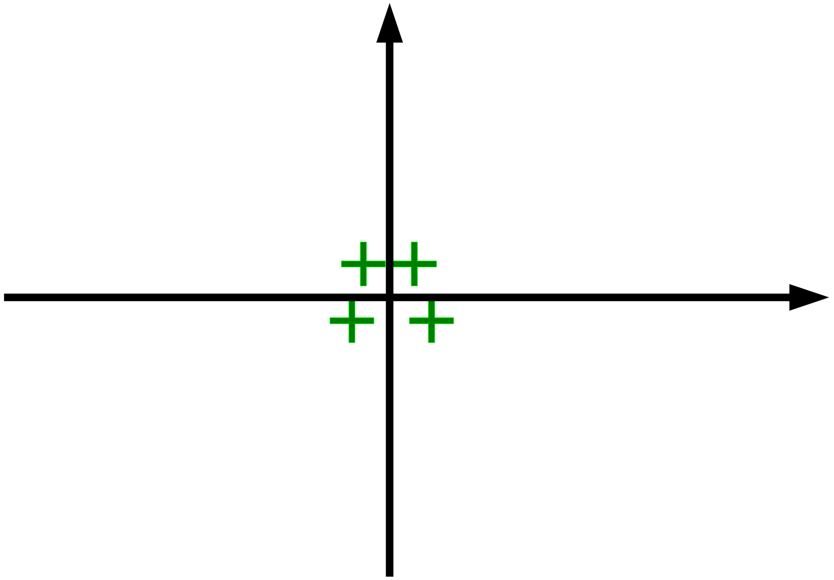


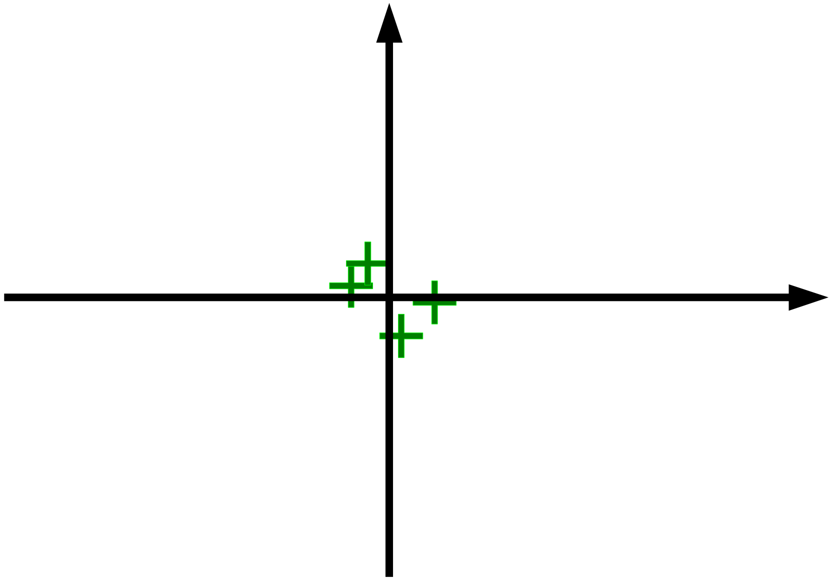
Consider them as vectors

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using singular value decomposition. Compute the two necessary homography matrices.





1. Conditioning

- Translate the centroid of all points to the origin
- Scale mean distance to origin to one

$$T=\begin{bmatrix} \frac{1}{s_x} & 0 & \frac{-t_x}{s_x} \\ 0 & \frac{1}{s_y} & \frac{-t_y}{s_y} \\ 0 & 0 & 1 \end{bmatrix}$$

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using singular value decomposition. Compute the two necessary homography matrices.

2. Create design matrix

$$A_i = \begin{bmatrix} -\tilde{w}'\tilde{\mathbf{x}}_i^T & \mathbf{0} & \tilde{u}_i'\tilde{\mathbf{x}}_i^T \\ \mathbf{0} & -\tilde{w}'\tilde{\mathbf{x}}_i^T & \tilde{v}_i'\tilde{\mathbf{x}}_i^T \end{bmatrix}$$

2. Exercise

$$\boxed{x' = H x}$$

Homography equation for points

(Note: for simplicity uncond. points are used)

2. Exercise

$$\boldsymbol{x}' = \boldsymbol{H}\,\boldsymbol{x}$$

Homography equation for points
(Note: for simplicity uncond. points are used)

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Use definitions of $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{H}$

2. Exercise

$$\boldsymbol{x}' = \boldsymbol{H}\,\boldsymbol{x}$$

Homography equation for points
(Note: for simplicity uncond. points are used)

Use definitions of $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{H}$

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Write as linear equation system

$$\begin{aligned} u' &= h_{11} u + h_{12} v + h_{13} w \\ v' &= h_{21} u + h_{22} v + h_{23} w \\ w' &= h_{31} u + h_{32} v + h_{33} w \end{aligned}$$

2. Exercise

$$\boldsymbol{x}' = \boldsymbol{H}\,\boldsymbol{x}$$

Homography equation for points
(Note: for simplicity uncond. points are used)

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Use definitions of $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{H}$

$$\begin{aligned} u' &= h_{11}u + h_{12}v + h_{13}w \\ v' &= h_{21}u + h_{22}v + h_{23}w \\ w' &= h_{31}u + h_{32}v + h_{33}w \end{aligned}$$

Write as linear equation system

$$\frac{u'}{w'} = \frac{h_{11}u + h_{12}v + h_{13}w}{h_{31}u + h_{32}v + h_{33}w}$$

$$\frac{v'}{w'} = \frac{h_{21}u + h_{22}v + h_{23}w}{h_{31}u + h_{32}v + h_{33}w}$$

Use transformation from homogeneous to Euclidean coordinates

2. Exercise

$$u' \cdot (h_{31} u + h_{32} v + h_{33} w) = w' \cdot (h_{11} u + h_{12} v + h_{13} w)$$
$$v' \cdot (h_{31} u + h_{32} v + h_{33} w) = w' \cdot (h_{21} u + h_{22} v + h_{23} w)$$

Rewrite

2. Exercise

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{11}u+h_{12}v+h_{13}w)$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{21}u+h_{22}v+h_{23}w)$$

Rewrite

Rewrite

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{11}u+h_{12}v+h_{13}w)=0$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{21}u+h_{22}v+h_{23}w)=0$$

2. Exercise

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{11}u+h_{12}v+h_{13}w)$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{21}u+h_{22}v+h_{23}w)$$

Rewrite

Rewrite

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{11}u+h_{12}v+h_{13}w)=0$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{21}u+h_{22}v+h_{23}w)=0$$

Rewrite

$$u'h_{31}u+u'h_{32}v+u'h_{33}w-w'h_{11}u-w'h_{12}v-w'h_{13}w=0$$
$$v'h_{31}u+v'h_{32}v+v'h_{33}w-w'h_{21}u-w'h_{22}v-w'h_{23}w=0$$

2. Exercise

Rewrite

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{11}u+h_{12}v+h_{13}w)$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{21}u+h_{22}v+h_{23}w)$$

Rewrite

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{11}u+h_{12}v+h_{13}w)=0$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{21}u+h_{22}v+h_{23}w)=0$$

Rewrite

$$u'h_{31}u+u'h_{32}v+u'h_{33}w-w'h_{11}u-w'h_{12}v-w'h_{13}w=0$$
$$v'h_{31}u+v'h_{32}v+v'h_{33}w-w'h_{21}u-w'h_{22}v-w'h_{23}w=0$$

Sort h_{ij}

$$-w'h_{11}u-w'h_{12}v-w'h_{13}w+u'h_{31}u+u'h_{32}v+u'h_{33}w=0$$
$$-w'h_{21}u-w'h_{22}v-w'h_{23}w+v'h_{31}u+v'h_{32}v+v'h_{33}w=0$$

2. Exercise

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{11}u+h_{12}v+h_{13}w)$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)=w'\cdot(h_{21}u+h_{22}v+h_{23}w)$$

Rewrite

Rewrite

$$u'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{11}u+h_{12}v+h_{13}w)=0$$
$$v'\cdot(h_{31}u+h_{32}v+h_{33}w)-w'\cdot(h_{21}u+h_{22}v+h_{23}w)=0$$

Rewrite

$$u'h_{31}u+u'h_{32}v+u'h_{33}w-w'h_{11}u-w'h_{12}v-w'h_{13}w=0$$
$$v'h_{31}u+v'h_{32}v+v'h_{33}w-w'h_{21}u-w'h_{22}v-w'h_{23}w=0$$

Sort h_{ij}

$$-w'h_{11}u-w'h_{12}v-w'h_{13}w+u'h_{31}u+u'h_{32}v+u'h_{33}w=0$$
$$-w'h_{21}u-w'h_{22}v-w'h_{23}w+v'h_{31}u+v'h_{32}v+v'h_{33}w=0$$

$$-w'h_{11}u-w'h_{12}v-w'h_{13}w+0\cdot h_{21}+0\cdot h_{22}+0\cdot h_{23}$$
$$+u'h_{31}u+u'h_{32}v+u'h_{33}w=0$$
$$0\cdot h_{11}+0\cdot h_{12}+0\cdot h_{13}-w'h_{21}u-w'h_{22}v-w'h_{23}w$$
$$+v'h_{31}u+v'h_{32}v+v'h_{33}w=0$$

Insert missing h_{ij} with 0-coefficients

2. Exercise

$$\begin{bmatrix} -w'u & -w'v & -w'w & 0 & 0 & 0 & u'u & u'v & u'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & -w'u & -w'v & -w'w & v'u & v'v & v'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Write as vector

2. Exercise

$$\begin{bmatrix} -w'u & -w'v & -w'w & 0 & 0 & 0 & u'u & u'v & u'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & -w'u & -w'v & -w'w & v'u & v'v & v'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Write as vector

Use definition of $\boldsymbol{x}$

$$\begin{bmatrix} -w'\boldsymbol{x}^T & \boldsymbol{0} & u'\boldsymbol{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} \boldsymbol{0} & -w'\boldsymbol{x}^T & v'\boldsymbol{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

2. Exercise

$$\begin{bmatrix} -w'u & -w'v & -w'w & 0 & 0 & 0 & u'u & u'v & u'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & -w'u & -w'v & -w'w & v'u & v'v & v'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Write as vector

Use definition of x

$$\begin{bmatrix} -w'\mathbf{x}^T & \mathbf{0} & u'\mathbf{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} \mathbf{0} & -w'\mathbf{x}^T & v'\mathbf{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Write as matrix

$$\begin{bmatrix} -w'\mathbf{x}^T & \mathbf{0} & u'\mathbf{x}^T \\ \mathbf{0} & -w'\mathbf{x}^T & v'\mathbf{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

2. Exercise

$$\begin{bmatrix} -w'u & -w'v & -w'w & 0 & 0 & 0 & u'u & u'v & u'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & -w'u & -w'v & -w'w & v'u & v'v & v'w \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Write as vector

Use definition of x

$$\begin{bmatrix} -w'\boldsymbol{x}^T & \boldsymbol{0} & u'\boldsymbol{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

$$\begin{bmatrix} \boldsymbol{0} & -w'\boldsymbol{x}^T & v'\boldsymbol{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Write as matrix

$$\begin{bmatrix} -w'\boldsymbol{x}^T & \boldsymbol{0} & u'\boldsymbol{x}^T \\ \boldsymbol{0} & -w'\boldsymbol{x}^T & v'\boldsymbol{x}^T \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^T = 0$$

Homogeneous equation system

$$Ah = 0$$

2. Exercise

A corresponding homogeneous equation system $A_i\, h = 0$ can be derived for every point pair $\left(\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i'\right)$.

For numerical reasons conditioned points $\left(\tilde{\boldsymbol{x}}_i \leftrightarrow \tilde{\boldsymbol{x}}_i'\right)$ are used.

Therefore:

$$A_i = \begin{bmatrix} -\tilde{w}_i' \tilde{\mathbf{x}}_i^T & \mathbf{0} & \tilde{u}_i' \tilde{\mathbf{x}}_i^T \\ \mathbf{0} & -\tilde{w}_i' \tilde{\mathbf{x}}_i^T & \tilde{v}_i' \tilde{\mathbf{x}}_i^T \end{bmatrix} \in \mathbb{R}^{2,9}$$

2. Exercise

A corresponding homogeneous equation system $A_i h = 0$ can be derived for every point pair $(\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i')$.

For numerical reasons conditioned points $(\tilde{\boldsymbol{x}}_i \leftrightarrow \tilde{\boldsymbol{x}}_i')$ are used.

Therefore:

$$A_i = \begin{bmatrix} -\tilde{w}_i' \tilde{\boldsymbol{x}}_i^T & \boldsymbol{0} & \tilde{u}_i' \tilde{\boldsymbol{x}}_i^T \\ \boldsymbol{0} & -\tilde{w}_i' \tilde{\boldsymbol{x}}_i^T & \tilde{v}_i' \tilde{\boldsymbol{x}}_i^T \end{bmatrix} \in \mathbb{R}^{2,9}$$

And for N point pairs:

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_N \end{bmatrix} \in \mathbb{R}^{2\mathrm{N},9}$$

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using the singular value decomposition. Compute the two necessary homography matrices.

2. Create design matrix

$$A_i = \begin{bmatrix} -\tilde{w}'\tilde{\boldsymbol{x}}_i^T & \boldsymbol{0} & \tilde{u}_i'\tilde{\boldsymbol{x}}_i^T \\ \boldsymbol{0} & -\tilde{w}'\tilde{\boldsymbol{x}}_i^T & \tilde{v}_i'\tilde{\boldsymbol{x}}_i^T \end{bmatrix}$$

3. Solve equation system with SVD

$$\boldsymbol{A}\boldsymbol{h} = 0 \qquad \boldsymbol{h} = (a, b, c, d, e, f, g, h, j)^T$$

2. Exercise

3. Homography Computation:

Implement C++-function to estimate a 2D homography using the singular value decomposition. Compute the two necessary homography matrices.

2. Create design matrix

$$A_i = \begin{bmatrix} -\tilde{w}'\,\tilde{\boldsymbol{x}}_i^T & \boldsymbol{0} & \tilde{u}_i'\,\tilde{\boldsymbol{x}}_i^T \\ \boldsymbol{0} & -\tilde{w}'\,\tilde{\boldsymbol{x}}_i^T & \tilde{v}_i'\,\tilde{\boldsymbol{x}}_i^T \end{bmatrix}$$

3. Solve equation system with SVD

$$\boldsymbol{A}\boldsymbol{h} = 0 \qquad \boldsymbol{h} = (a, b, c, d, e, f, g, h, j)^T$$

4. Reshape and deconditioning

$$\tilde{\boldsymbol{H}} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & j \end{bmatrix} \qquad \boldsymbol{H} = \boldsymbol{T}'^{-1}\,\tilde{\boldsymbol{H}}\,\boldsymbol{T}$$

2. Exercise

4. Projective Rectification:

Use the estimated homographies to adapt the first and the third image to the second one.

2. Exercise

4. Projective Rectification:

Use the estimated homographies to adapt the first and the third image to the second one.

5. Visualization

Show the produced panorama image on the screen and save it to disk.

2. Exercise

4. Projective Rectification:

Use the estimated homographies to adapt the first and the third image to the second one.

5. Visualization

Show the produced panorama image on the screen and save it to disk.

Optional:

Change code in order to
1) Process arbitrary number of images.
2) Find corresponding point pairs automatically.

2. Exercise – Part I : Theory

2D Homography

1. Which conditions on the image acquisition have to be fulfilled in order to model the image transformation successfully as a 2D homography ?

2. How many corresponding point pairs do you need to reconstruct a 2D homography?

3. Why should the homogeneous points be normalized before creating the design matrix?

4. What are possible reasons why the images don't align perfectly align?

2. Exercise – Part II : Practical

2D Homography

Given:

- Main function
  - Variable declaration
  - Call of necessary functions
- Header of individual functions
- Point selection
- Stitching function

Todo:

- Individual functions
  - Fill in the necessary function body parts
- Take three pictures
- Process them to a panorama
  - Include input pictures and output pictures in the submission

2. Exercise – Given

```
int getPoints(Mat& baseImg, Mat& attachImg,
              vector<Vec3f>& p_base,
              vector<Vec3f>& p_attach)
```

`baseImg`: Image data of first (base) image
`attachImg`: Image data of second image
`p_base`: Output, points in first image (homogeneous coordinates)
`p_attach`: Output, points in second image (homogeneous coordinates)

- Displays the two images in two individual windows.
- Catches left mouse clicks, stores position in corresponding array.
- Marks points in images by green circles.

2. Exercise – Given

`Mat stitch(Mat& base, Mat& attach, Matx33f& H)`

`base`:	Image data and window title of first (base) image
`attach`:	Image data and window title of second image
`H`:	Computed homography that relates second to base image
`return`:	The generated panorama image

- Takes two images and a corresponding homography
- Warps the second image according to the given transformation
- Copies both images into a single output image

2. Exercise – Reuse

```
Vec3f eucl2hom_point_2D(Vec2f& p)
```

`p` : point in Euclidean coordinates
`return` : same point in homogeneous coordinates

- Reuse your implementation from exercise 1

```
vector<Vec3f> applyH_2D(vector<Vec3f>& geomObj,
                        Matx33f& H, GeometryType type)
```

`geomObj` : the geometric objects, that have to be transformed
`H` : the homography of the transformation
`type` : determines the type of the geometric object (point/line)
`return` : the transformed objects

- Reuse your implementation from exercise 1

$$x' = H \cdot x$$

2. Exercise – To Do

```
Matx33f getCondition2D(vector<Vec3f>& p)
```

`p`: N points as an array of 3D vectors
Homogeneous, but you may assume last component is 1

`return`: The 3x3 matrix for point conditioning

- Computes the transformation matrix to:
  - move centroid (mean) to origin
  - scale so mean absolute x-distance to origin is one
  - scale so mean absolute y-distance to origin is one

$$T = \begin{bmatrix} \frac{1}{s_x} & 0 & \frac{-t_x}{s_x} \\ 0 & \frac{1}{s_y} & \frac{-t_y}{s_y} \\ 0 & 0 & 1 \end{bmatrix}$$

2. Exercise – To Do

```
Matx33f getCondition2D(vector<Vec3f>& p)
```

`p`: N points as an array of 3D vectors
Homogeneous, but you may assume last component is 1

`return`: The 3x3 matrix for point conditioning

```
vector<Vec3f> applyH_2D(vector<Vec3f>& geomObj,
                        Matx33f& H, GeometryType type)
```

`geomObj`: N geometric objects as 3 x N matrix (here: unconditioned points)

`H`: 3x3 transformation matrix

`return`: N geometric objects as 3 x N matrix (here: conditioned points)

- Reuse your implementation from exercise 1

$$x' = H \cdot x$$

2. Exercise – To Do

```
Mat_<float> getDesignMatrix_homography2D(
                    vector<Vec3f>& base, vector<Vec3f>& attach)
```

`base`: N conditioned points as array of 3D vectors from first image
`attach`: N conditioned points as array of 3D vectors from second image
`return`: Designmatrix to compute 2D homography

- Takes corresponding point pairs (already conditioned)
- Generates design matrix to compute 2D homography
- Make it work for arbitrary numbers of point pairs

$$A_i = \begin{bmatrix} -\tilde{w}_i' \tilde{\mathbf{x}}_i^T & \mathbf{0} & \tilde{u}_i' \tilde{\mathbf{x}}_i^T \\ \mathbf{0} & -\tilde{w}_i' \tilde{\mathbf{x}}_i^T & \tilde{v}_i' \tilde{\mathbf{x}}_i^T \end{bmatrix}$$

2. Exercise – To Do

`Matx33f solve_dlt_homography2D(Mat_<float>& A)`

`A`: Design matrix representing homogeneous equation system Ah = 0
`return`: 2D homography

- Use SVD to solve homogeneous equation system
  - → `cv::SVD svd(A, SVD::FULL_UV)`
  - → `svd.u    Left singular vectors (not needed)`
  - → `svd.w    Singular values`
  - → `svd.vt   Transposed right singular values`
  - → Access: `svd.vt.at<float>(row, col)`
  - → cv::SVD already sorts the singular values by size

- Reshape result into 3x3 matrix

2. Exercise – To Do

```
Matx33f decondition_homography2D(Matx33f& T_base, Matx33f& T_attach, Matx33f& H)
```

`T_base`:	Condition matrix of first set of points (base)
`T_attach`:	Condition matrix of second set of points
`H`:	2D homography
returns:	Deconditioned 2D homography

- Deconditons 2D homography

$$\tilde{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & j \end{bmatrix} \qquad H = T'^{-1} \tilde{H} T$$

2. Exercise – To Do

```
Matx33f homography2D(vector<Vec3f>& base,
                     vector<Vec3f>& attach)
```

`base`: N points as array of 3D vectors from first image
`attach`: N points as array of 3D vectors from second image
`return`: Output, 2D homography

- Computes 2D homography from given set of point pairs
- Calls:

```
2x getCondition2D(...)
2x applyH_2D(...)
getDesignMatrix_homography2D(...)
solve_dlt_homography2D(...)
decondition_homography2D(...)
```

Deadlines and Next Meeting

Deadline: In 2 weeks
15.12.2020, 10:00