Bauhaus-
Universität
Weimar

Photogrammetric Computer Vision

Exercise 2

Winter semester 21/22

(Course materials for internal use only!)

Computer Vision in Engineering – Prof. Dr. Rodehorst
M.Sc. Mariya Kaisheva
mariya.kaisheva@uni-weimar.de

Agenda

Topics:

Assignment 1.	Points and lines in the plane, first steps in MATLAB / Octave
Assignment 2.	**Projective transformation (Homography)**
Assignment 3.	Camera calibration using direct linear transformation (DLT)
Assignment 4.	Orientation of an image pair
Assignment 5.	Projective and direct Euclidean reconstruction
Assignment 6.	Stereo image matching
Final Project. *	- will be announced later -

***Depending on the regulations of your study program, this project might be optional for you!**

If you are not sure about the exact requirements for your study program,
please consult with a representative of the Academic Affairs Office in charge!

Agenda

	Beginning:	Submission deadline:
Assignment 1.	~~18.10.21~~	~~31.10.21~~
Assignment 2.	**01.11.21**	**14.11.21**
Assignment 3.	15.11.21	28.11.21
Assignment 4.	29.11.21	12.12.21
Assignment 5.	13.12.21	09.01.22
Assignment 6.	10.01.22	23.01.22
Final Project. *	24.01.22	- will be announced later -

***Depending on the regulations of your study program, this project might be optional for you!**

If you are not sure about the exact requirements for your study program, please consult with a representative of the Academic Affairs Office in charge!

Bauhaus-Universität Weimar

Quick Question on…

2D Transformations & Geometric Spaces

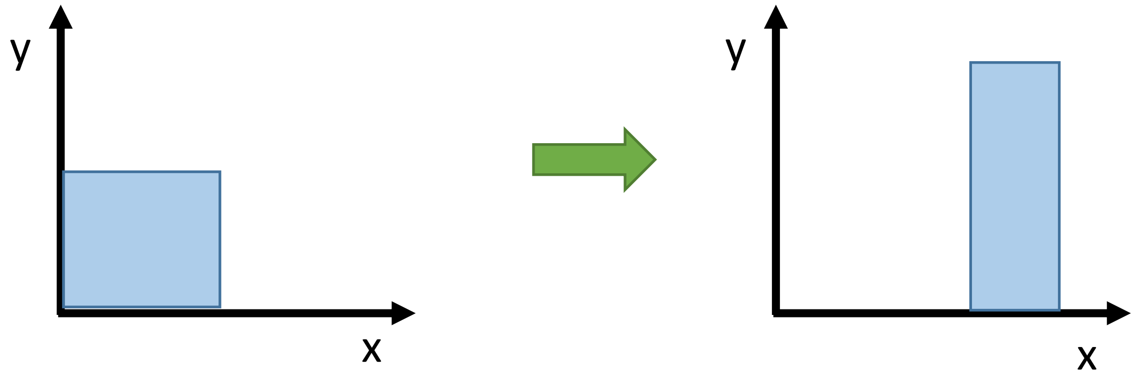
What geometric space allows the following 2D transformation?

A. Projective

B. Metric

C. Affine

D. Euclidian



Assignment 1 – sample solution

Part 1

1. You would like to compute the connecting line between two 2D points. What happens, if the two points are identical?

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ w \end{pmatrix}, \quad \mathbf{l} = \mathbf{x} \times \mathbf{x} = \begin{pmatrix} yw - wy \\ wx - xw \\ xy - yx \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad |\mathbf{l}| = 0$$

Not defined!

2. Where does the general line $x \cos \varphi + y \sin \varphi = d$ intersect the line $(0, 0, 1)^T$ given in homogeneous coordinates? How can this point be interpreted?

$$\mathbf{l}_1 = \begin{pmatrix} \cos\phi \\ \sin\phi \\ -d \end{pmatrix}, \quad \mathbf{l}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2 = \begin{pmatrix} \sin\phi \\ -\cos\phi \\ 0 \end{pmatrix}$$

Section at infinity in the direction l_1

3. Show that the horizon is a straight line by showing that three points on the horizon are always collinear.

$$\det\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 0 & 0 & 0 \end{pmatrix} = x_1 \cdot y_2 \cdot 0 - x_3 \cdot y_2 \cdot 0 + x_2 \cdot y_3 \cdot 0 - x_1 \cdot y_3 \cdot 0 + x_3 \cdot y_1 \cdot 0 - x_2 \cdot y_1 \cdot 0 = 0$$

Part 2

All objects in MATLAB are matrices. A matrix is created with `[1,2;3,4]`, where semicolons separate the rows. For a matrix multiplication use * and for a matrix `A` is `A'` the transpose. For the solution of this exercise the commands `cross`, `sin`, `cos`, `pi` and `inv` can be helpful.

1. The two points $\mathbf{x} = (2, 3)^T$ and $\mathbf{y} = (-4, 5)^T$ are given.
   a. Determine the connecting line $\mathbf{l}$ between the two points.
   b. *Move* $\mathbf{x}$ and $\mathbf{y}$ in the direction $\mathbf{t} = (6, -7)^T$,
      rotate afterwards using the angle $\varphi = 15°$ and finally
      scale with factor $\lambda = 8$.
   c. Accomplish the same operations with the line $\mathbf{l}$.
2. Examine whether the transformed points $\mathbf{x}'$ and $\mathbf{y}'$ are on the transformed line $\mathbf{l}'$.

Part 2

$$\mathbf{l}^{\mathsf{T}}\mathbf{x} = 0$$

$$\mathbf{l}'^{\mathsf{T}}\mathbf{x}' = 0$$

```
function Exercise1                        % Planar similarity transformation
%        =========
x = [ 2; 3; 1];                      % Points x and y in homogeneous coordinates
y = [-4; 5; 1];
l = cross(x, y)                            % Joining line l using cross product

H = Scale(8) * Rot(15) * Trans(6,-7);          % Transformation concatenation
                                 % 1. Translation, 2. Rotation, 3. Global scaling
x2 = H * x                               % Apply transformation to points x and y
y2 = H * y
l2 = inv(H') * l;                                 % Apply transformation to line l

x2' * l2                                    % Incidence test: scalar product 0?
y2' * l2

function T = Trans(x,y)
%        ==============
T = [ 1  0  x ;                                             % Translation matrix
      0  1  y ;
      0  0  1 ];

function R = Rot(a)
%        ==========
phi = a * pi / 180;                                         % Degree -> radiant
R = [ cos(phi) -sin(phi)  0 ;                                % Rotation matrix
      sin(phi)  cos(phi)  0 ;
         0         0      1 ];

function S = Scale(s)
%        ============
S = [ s  0  0 ;                                          % Global scaling matrix
      0  s  0 ;
      0  0  1 ];
```

Part 2

$$\mathbf{l}^{\mathrm{T}}\mathbf{x} = 0$$
$$\mathbf{l}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{H}\mathbf{x} = 0$$

$$\mathbf{l}'^{\mathrm{T}}\mathbf{x}' = 0$$

```
function Exercise1                          % Planar similarity transformation
%        =========
x = [ 2; 3; 1];                      % Points x and y in homogeneous coordinates
y = [-4; 5; 1];
l = cross(x, y)                              % Joining line l using cross product

H = Scale(8) * Rot(15) * Trans(6,-7);              % Transformation concatenation
                              % 1. Translation, 2. Rotation, 3. Global scaling
x2 = H * x                                 % Apply transformation to points x and y
y2 = H * y
l2 = inv(H') * l;                                  % Apply transformation to line l

x2' * l2                                        % Incidence test: scalar product 0?
y2' * l2

function T = Trans(x,y)
%        ==============
T = [ 1  0  x ;                                               % Translation matrix
      0  1  y ;
      0  0  1 ];

function R = Rot(a)
%        ==========
phi = a * pi / 180;                                              % Degree -> radiant
R = [ cos(phi) -sin(phi)  0 ;                                    % Rotation matrix
      sin(phi)  cos(phi)  0 ;
         0         0      1 ];

function S = Scale(s)
%        ============
S = [ s  0  0 ;                                             % Global scaling matrix
      0  s  0 ;
      0  0  1 ];
```

Part 2

$$\mathbf{l}^{\mathrm{T}}\mathbf{x} = 0$$

$$\mathbf{l}^{\mathrm{T}}\mathbf{H}^{-1}\boxed{\mathbf{Hx}} = 0$$

$$\mathbf{Hx} = \mathbf{x}'$$

$$\mathbf{l}'^{\mathrm{T}}\mathbf{x}' = 0$$

```
function Exercise1                              % Planar similarity transformation
%        =========
x = [ 2; 3; 1];                        % Points x and y in homogeneous coordinates
y = [-4; 5; 1];
l = cross(x, y)                                % Joining line l using cross product

H = Scale(8) * Rot(15) * Trans(6,-7);          % Transformation concatenation
                                   % 1. Translation, 2. Rotation, 3. Global scaling
x2 = H * x                                   % Apply transformation to points x and y
y2 = H * y
l2 = inv(H') * l;                                 % Apply transformation to line l

x2' * l2                                        % Incidence test: scalar product 0?
y2' * l2

function T = Trans(x,y)
%        ==============
T = [ 1   0   x ;                                              % Translation matrix
      0   1   y ;
      0   0   1 ];

function R = Rot(a)
%        ==========
phi = a * pi / 180;                                            % Degree -> radiant
R = [ cos(phi) -sin(phi)   0 ;                                   % Rotation matrix
      sin(phi)  cos(phi)   0 ;
         0         0       1 ];

function S = Scale(s)
%        ============
S = [ s   0   0 ;                                          % Global scaling matrix
      0   s   0 ;
      0   0   1 ];
```

Part 2

$$\mathbf{l}^{\mathsf{T}}\mathbf{x} = 0$$

$$\boxed{\mathbf{l}^{\mathsf{T}}\mathbf{H}^{-1}}\boxed{\mathbf{H}\mathbf{x}} = 0$$

$$\mathbf{l}^{\mathsf{T}}\mathbf{H}^{-1} = \mathbf{l}'^{\mathsf{T}} \qquad \mathbf{H}\mathbf{x} = \mathbf{x}'$$

$$\mathbf{l}'^{\mathsf{T}}\mathbf{x}' = 0$$

```
function Exercise1                                 % Planar similarity transformation
%        =========
x = [ 2; 3; 1];                           % Points x and y in homogeneous coordinates
y = [-4; 5; 1];
l = cross(x, y)                                  % Joining line l using cross product

H = Scale(8) * Rot(15) * Trans(6,-7);                  % Transformation concatenation
                                       % 1. Translation, 2. Rotation, 3. Global scaling
x2 = H * x                                    % Apply transformation to points x and y
y2 = H * y
l2 = inv(H') * l;                                      % Apply transformation to line l

x2' * l2                                             % Incidence test: scalar product 0?
y2' * l2

function T = Trans(x,y)
%        ==============
T = [ 1  0  x ;                                                      % Translation matrix
      0  1  y ;
      0  0  1 ];

function R = Rot(a)
%        ==========
phi = a * pi / 180;                                                 % Degree -> radiant
R = [ cos(phi) -sin(phi)  0 ;                                        % Rotation matrix
      sin(phi)  cos(phi)  0 ;
         0         0      1 ];

function S = Scale(s)
%        ============
S = [ s  0  0 ;                                                  % Global scaling matrix
      0  s  0 ;
      0  0  1 ];
```

Part 2

$$\mathbf{l}^\mathrm{T}\mathbf{x} = 0$$

$$\boxed{\mathbf{l}^\mathrm{T}\mathbf{H}^{-1}}\boxed{\mathbf{H}\mathbf{x}} = 0$$

$$\mathbf{l}^\mathrm{T}\mathbf{H}^{-1} = \mathbf{l}'^\mathrm{T} \qquad \mathbf{H}\mathbf{x} = \mathbf{x}'$$

$$\mathbf{H}^{-\mathrm{T}}\mathbf{l} = \mathbf{l}'$$

$$\mathbf{l}'^\mathrm{T}\mathbf{x}' = 0$$

```
function Exercise1                          % Planar similarity transformation
%        =========
x = [ 2; 3; 1];                    % Points x and y in homogeneous coordinates
y = [-4; 5; 1];
l = cross(x, y)                             % Joining line l using cross product

H = Scale(8) * Rot(15) * Trans(6,-7);           % Transformation concatenation
                                 % 1. Translation, 2. Rotation, 3. Global scaling
x2 = H * x                                % Apply transformation to points x and y
y2 = H * y
l2 = inv(H') * l;                              % Apply transformation to line l

x2' * l2                                     % Incidence test: scalar product 0?
y2' * l2

function T = Trans(x,y)
%        ==============
T = [ 1  0  x ;                                              % Translation matrix
      0  1  y ;
      0  0  1 ];

function R = Rot(a)
%        ==========
phi = a * pi / 180;                                          % Degree -> radiant
R = [ cos(phi) -sin(phi)  0 ;                                  % Rotation matrix
      sin(phi)  cos(phi)  0 ;
         0         0      1 ];

function S = Scale(s)
%        ============
S = [ s  0  0 ;                                           % Global scaling matrix
      0  s  0 ;
      0  0  1 ];
```

Assignment 2

Assignment 2: *Projective Transformation*

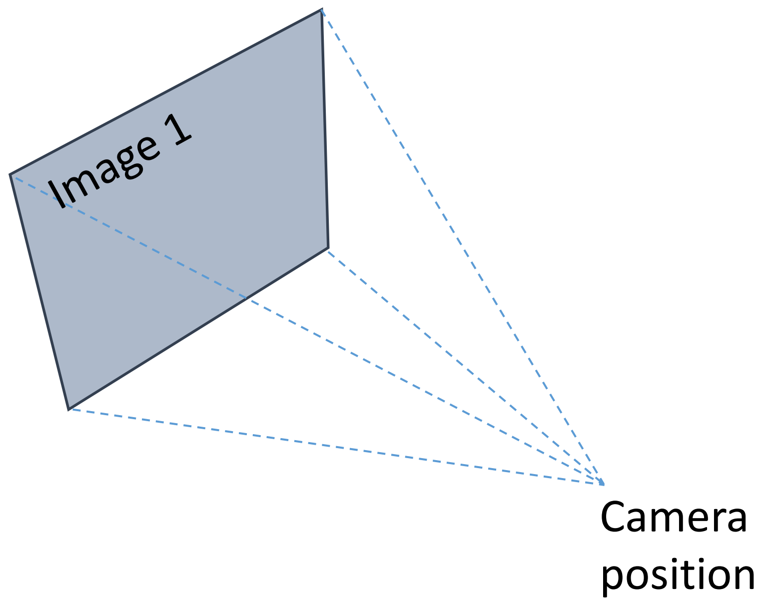
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - $\mathbf{H}_{12}$ (first to second image)
   - $\mathbf{H}_{32}$ (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Projective Transformation*

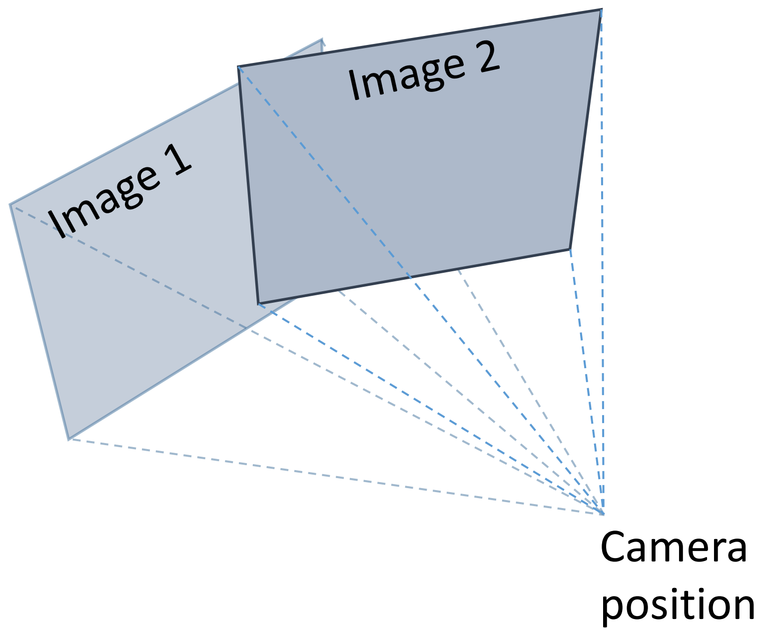
1) Image acquisition
    - 3 images
    - at least **30%** overlap

2) Correspondence analysis
    - **interactive** point selection

3) Homography computation
    - **H12** (first to second image)
    - **H32** (third image to intermediate mosaic)

4) Projective rectification
    - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Projective Transformation*

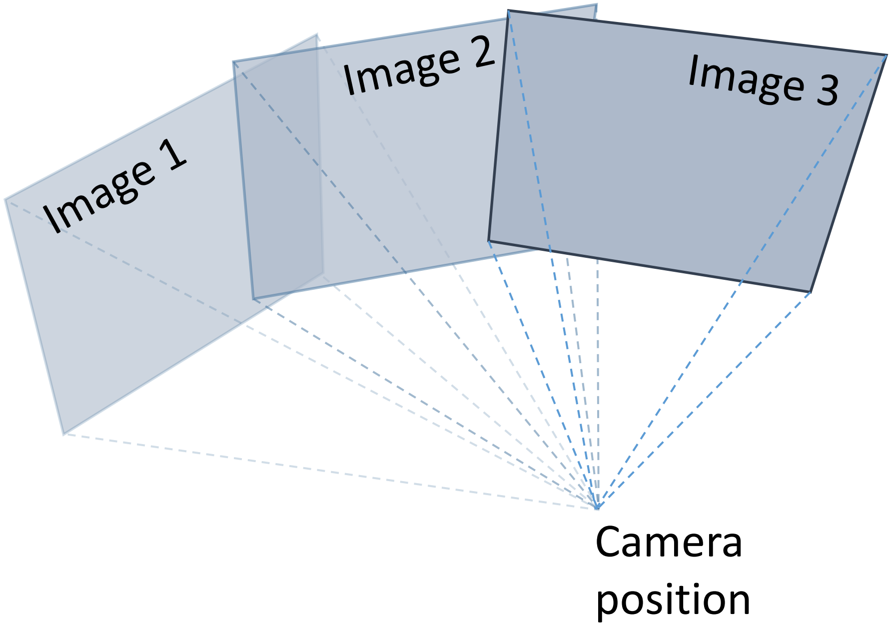
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - $\mathbf{H}_{12}$ (first to second image)
   - $\mathbf{H}_{32}$ (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Projective Transformation*

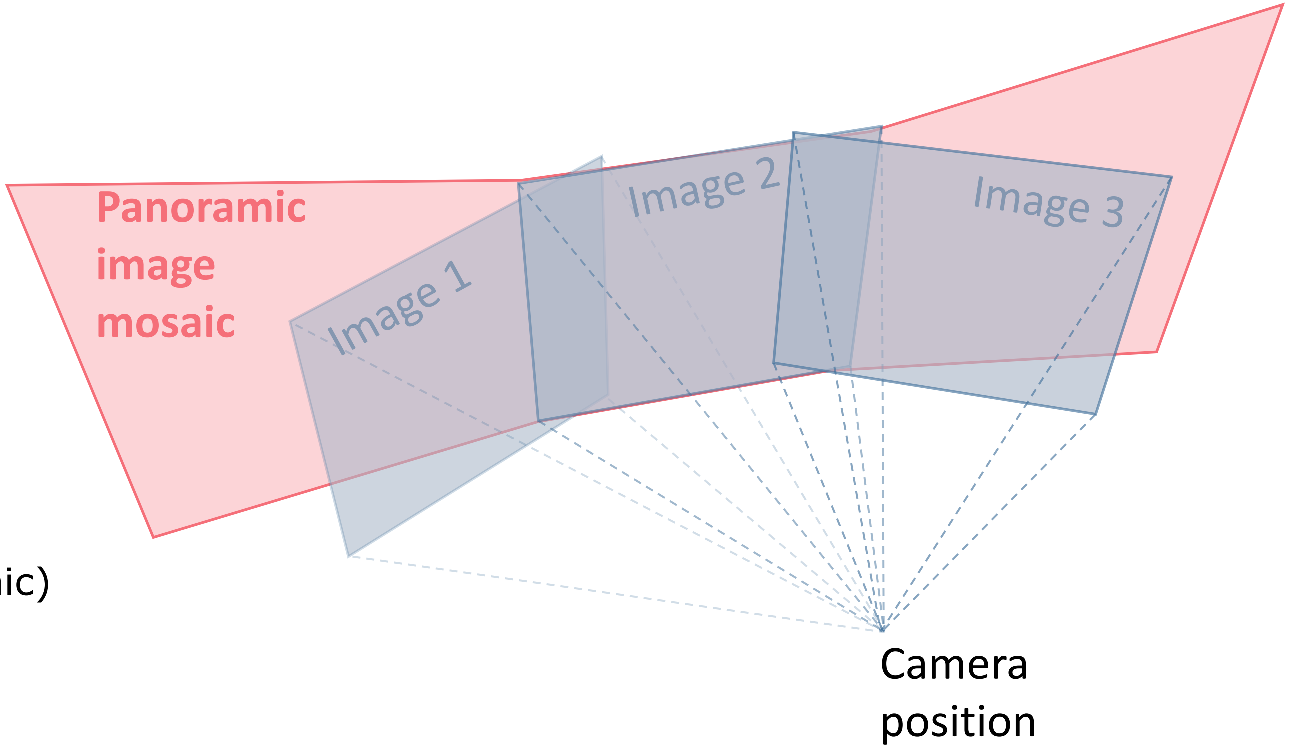
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - $\mathbf{H}_{12}$ (first to second image)
   - $\mathbf{H}_{32}$ (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Projective Transformation*

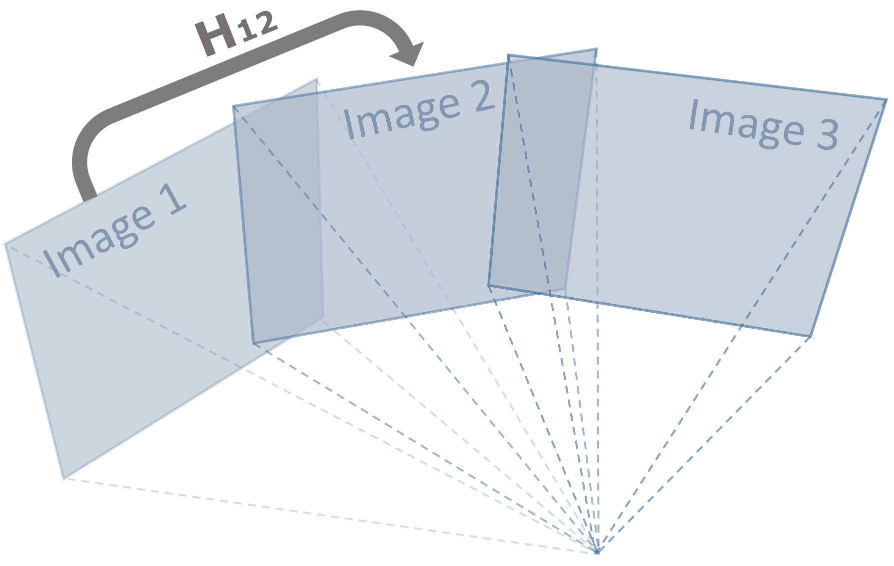
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - $\mathbf{H_{12}}$ (first to second image)
   - $\mathbf{H_{32}}$ (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Projective Transformation*

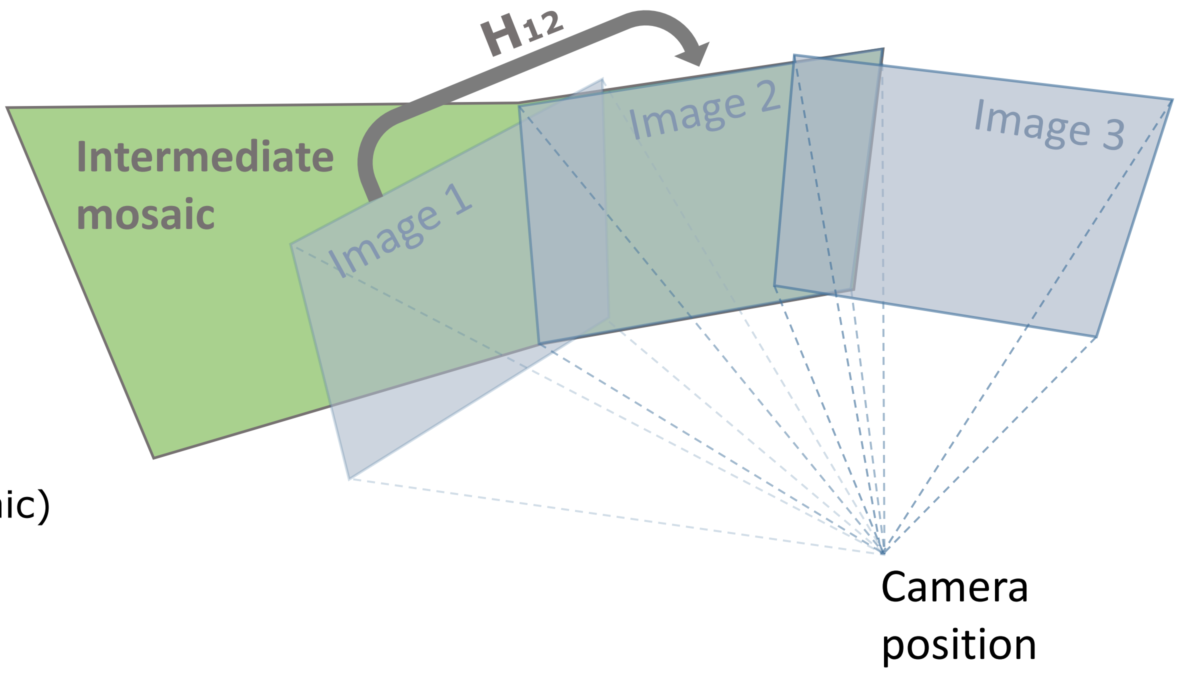
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - $\mathbf{H_{12}}$ (first to second image)
   - $\mathbf{H_{32}}$ (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Bauhaus-Universität Weimar

Assignment 2: *Projective Transformation*

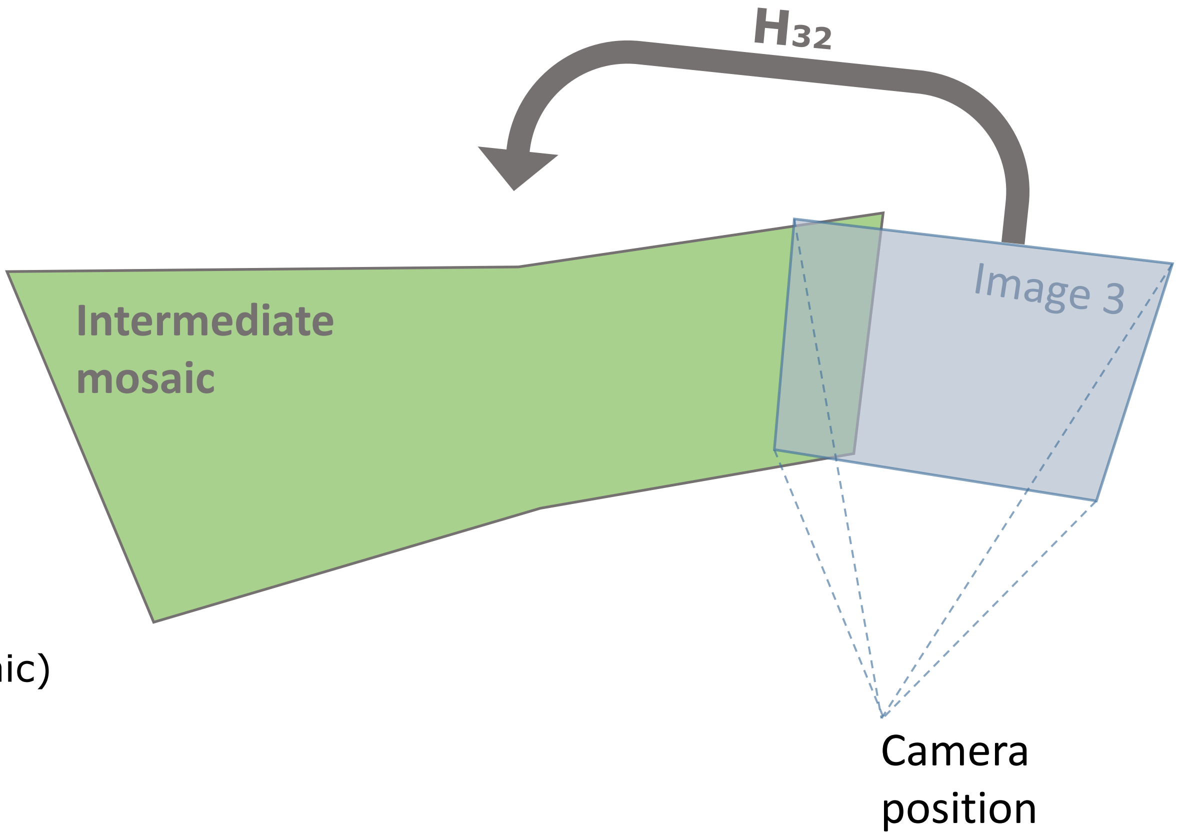
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - **H12** (first to second image)
   - **H32** (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Projective Transformation*

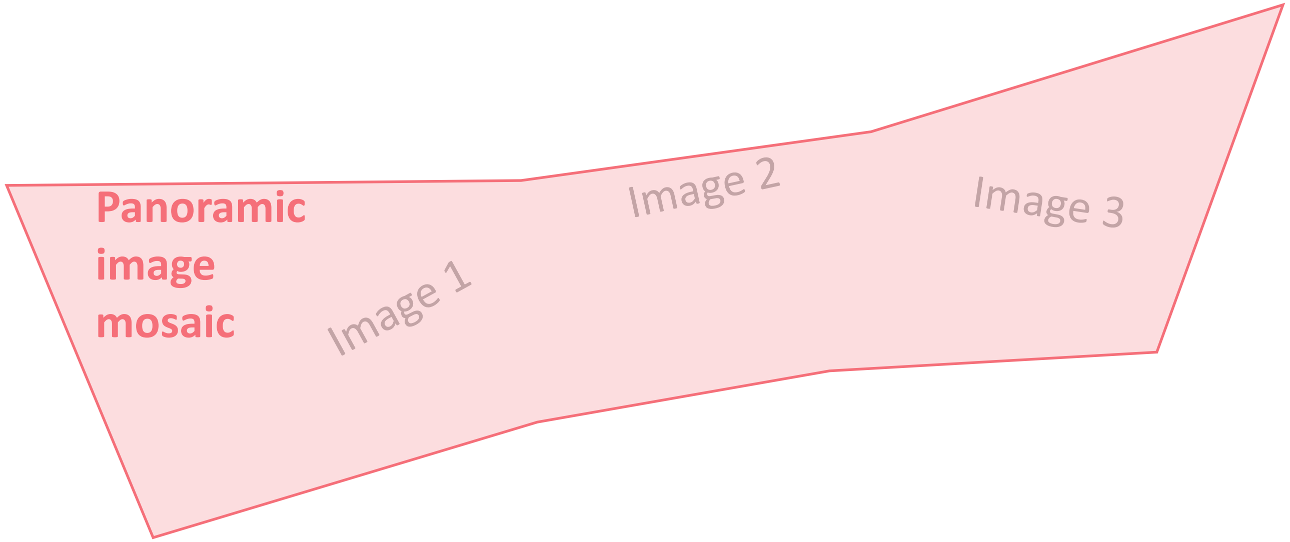
1) Image acquisition
   - 3 images
   - at least **30%** overlap

2) Correspondence analysis
   - **interactive** point selection

3) Homography computation
   - $\mathbf{H_{12}}$ (first to second image)
   - $\mathbf{H_{32}}$ (third image to intermediate mosaic)

4) Projective rectification
   - auxiliary program ***geokor***

5) Visualization



Assignment 2: *Example Result*

Image Source: *student submission; WS 19/20*