DMIP - Exercise:

Image Undistortion

Y. Xia, M. Bögel
Pattern Recognition Lab (CS 5)

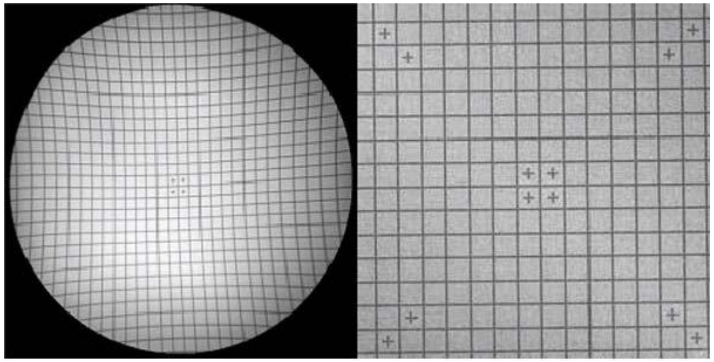
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Theory Review

Image Distortion

Images acquired from image intensifier (II) will suffer from distortion. This is mainly caused by:

- Earth magnetic field or artificial magnetic field
- Scattering
- A convex entrance screen



Siemens mobile C-arm (source: siemens)

Geometric Distortion Correction

Geometric distortion: the acquisition system modifies the geometry of the mapped object.

Correcting the geometric distortion needs a 3-step processing:

- Model design (parametric or non-parametric model, dimension of parameters and linear or non-linear estimator)
- Estimation of model parameters (Calibration): N points from undistorted image (x', y') and distorted image (x, y)

$$x_r = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} y_r'^j x_r'^i \qquad y_r = \sum_{i=0}^{d} \sum_{j=0}^{d-i} v_{ij} y_r'^j x_r'^i$$

- Inference → Interpolation of intensities of neighboring pixels

Pre-processing (create an artificial distorted image)

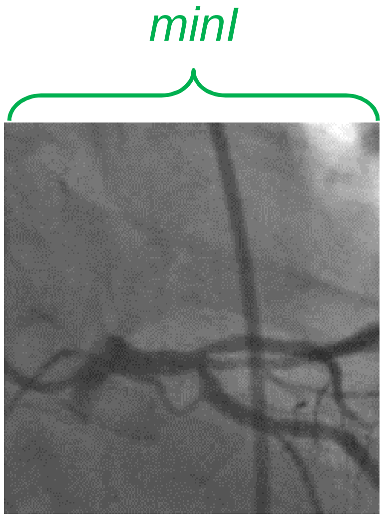
Pre-processing

Artificial Distortion Field

Pre-processing of input image

Create artificial distortion field

Bilinear interpolation



1. Generate a grid to sample the image

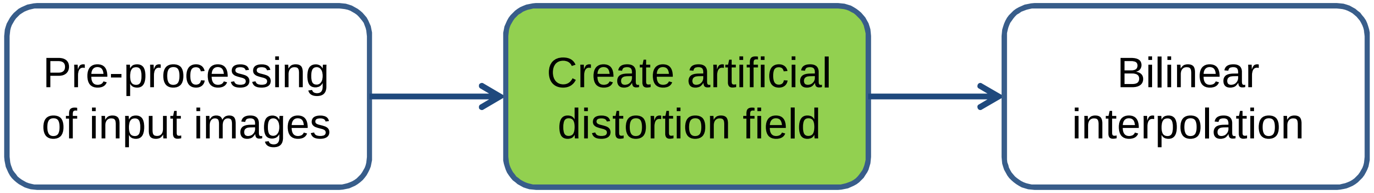
[X,Y] = meshgrid(1:minI,1:minI);

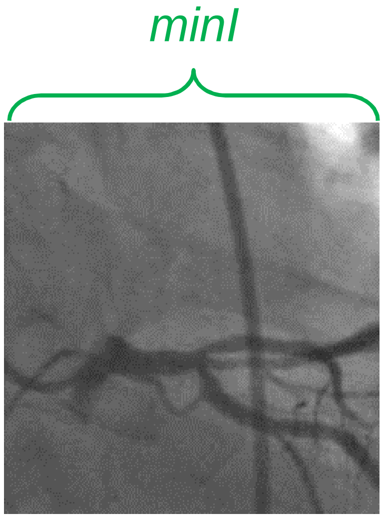
X = 1 2 3 4 …
1 2 3 4 …
1 2 3 4 …
1 2 3 4 …
.
.
.

Y = 1 1 1 1 …
2 2 2 2 …
3 3 3 3 …
4 4 4 4 …
.
.
.

Pre-processing

Artificial Distortion Field

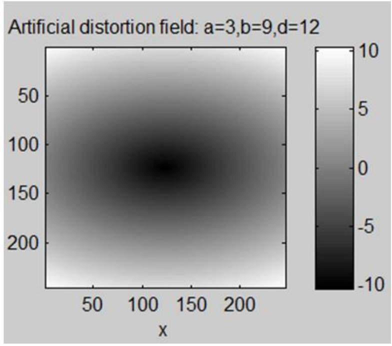




2. Create a distortion field (ellipsoidal)

$$R = d\sqrt{a\left(\frac{X-\frac{n}{2}}{n}\right)^2 + b\left(\frac{Y-\frac{m}{2}}{m}\right)^2}$$

where a: spread in x-direction
b: spread in y-direction
d: maximal value at the radius boundary



Resample Original Image

Pre-processing of input images

Create artificial distortion field

Bilinear interpolation

Resample the image *I* at new sample coordinates *(XD, YD)*

Idist = interp2(X,Y,I,XD,YD);

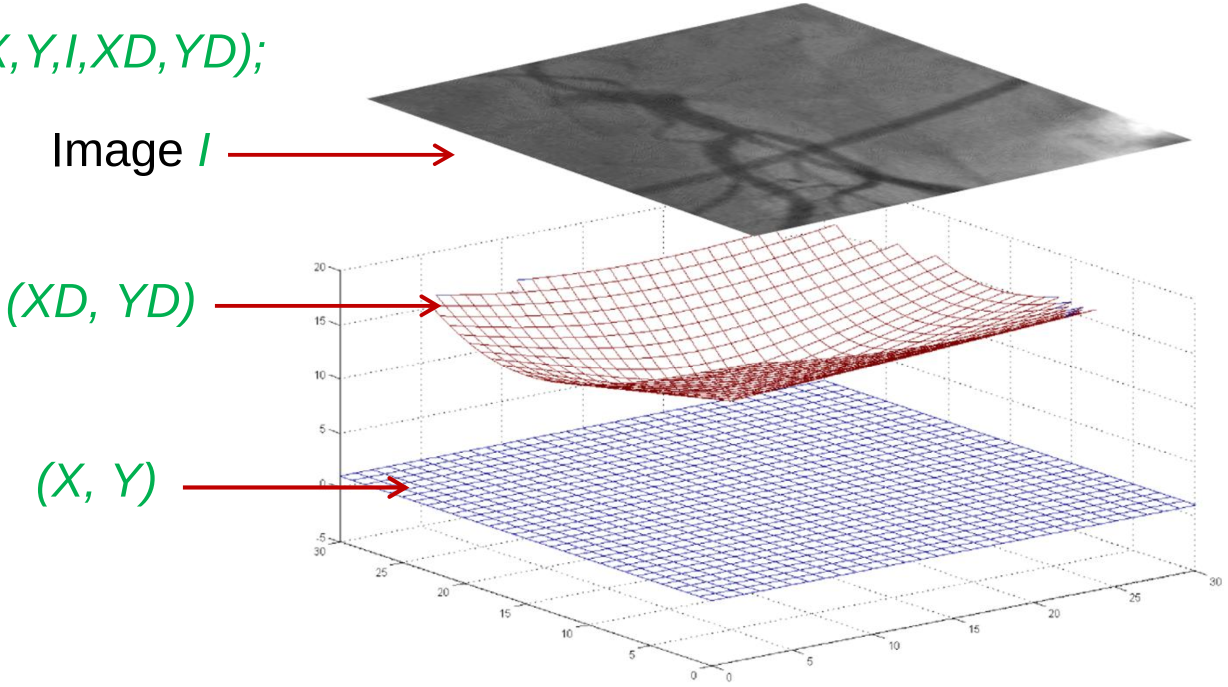
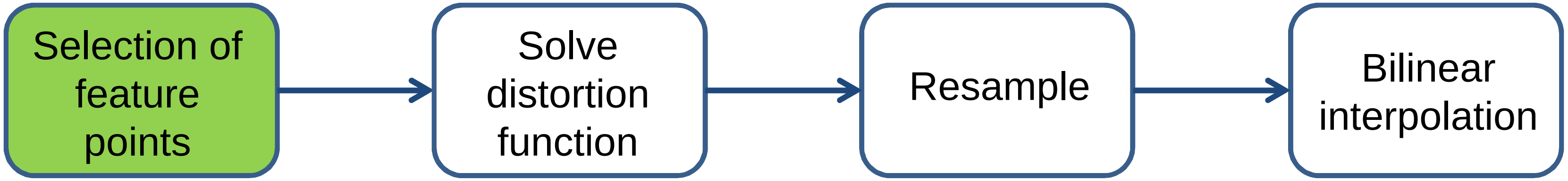


Image Undistortion - Workflow

Image Undistortion

Point Correspondences


Selection of feature points

Solve distortion function

Resample

Bilinear interpolation

In real world, we would know the relation between the undistorted and the distorted image by point correspondences of a calibration pattern.

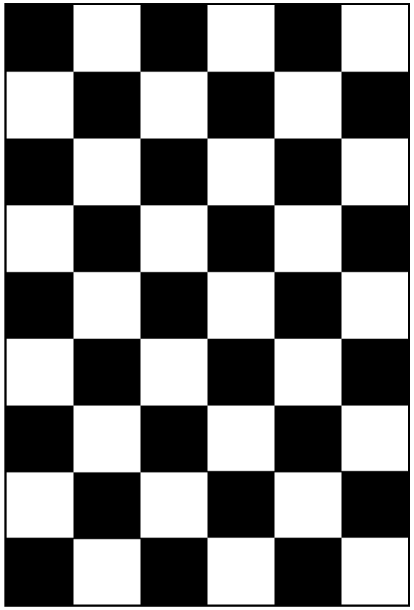
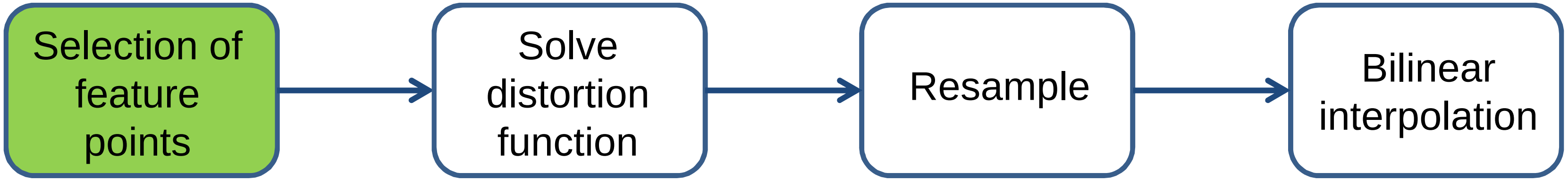
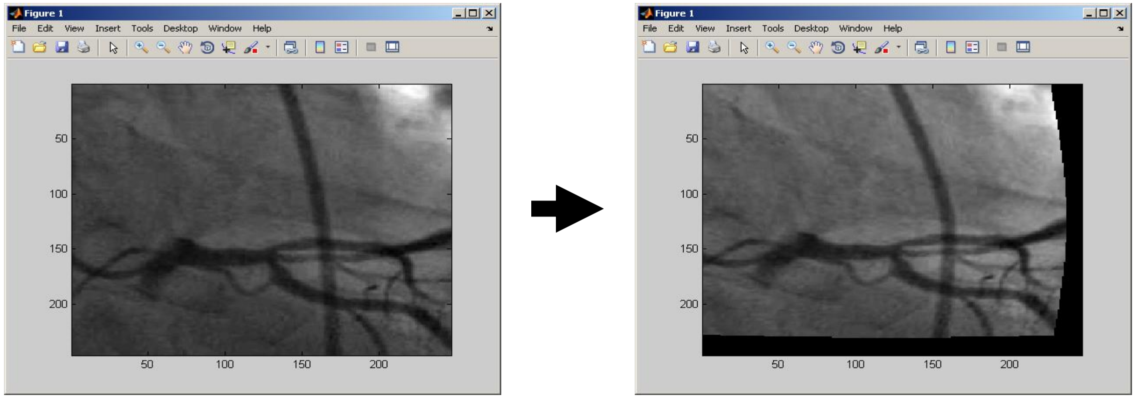


Image Undistortion

Choose Feature points



Here, we choose 8x8 lattice points (feature points) distributed over the whole image domain



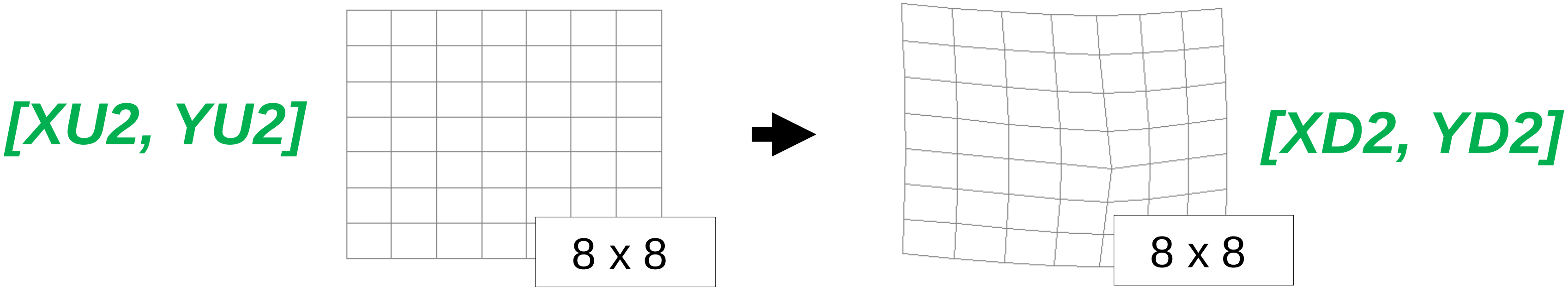
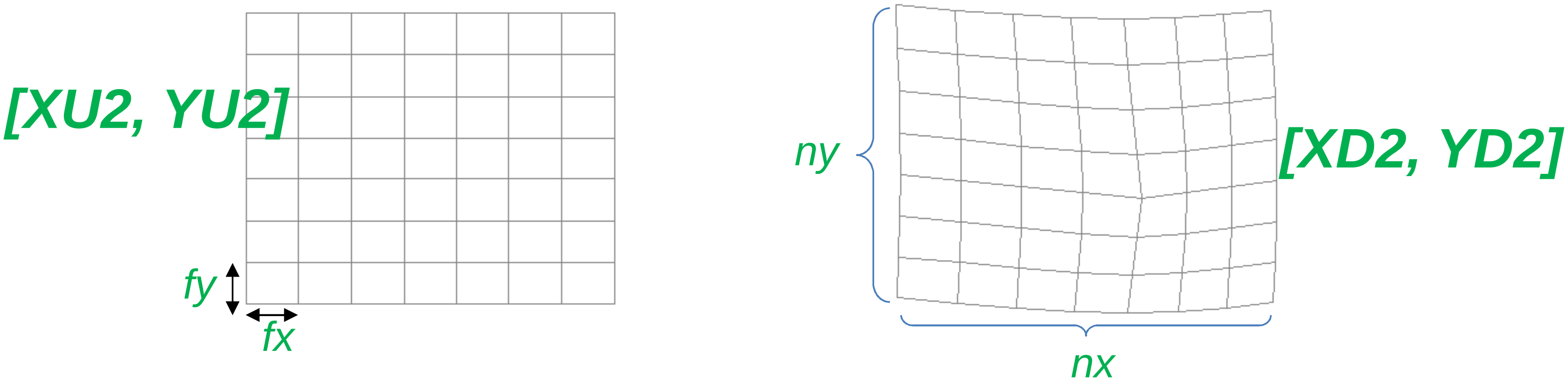


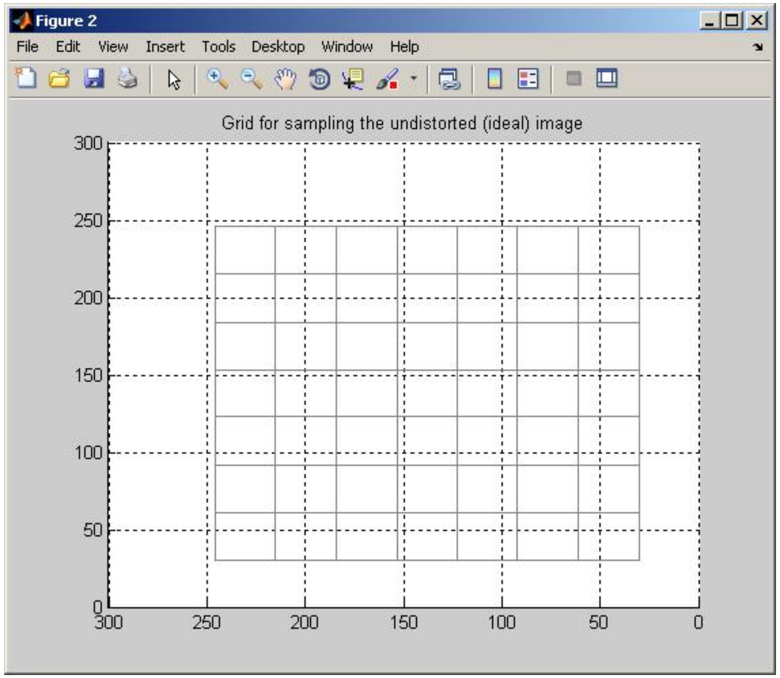
Image Undistortion

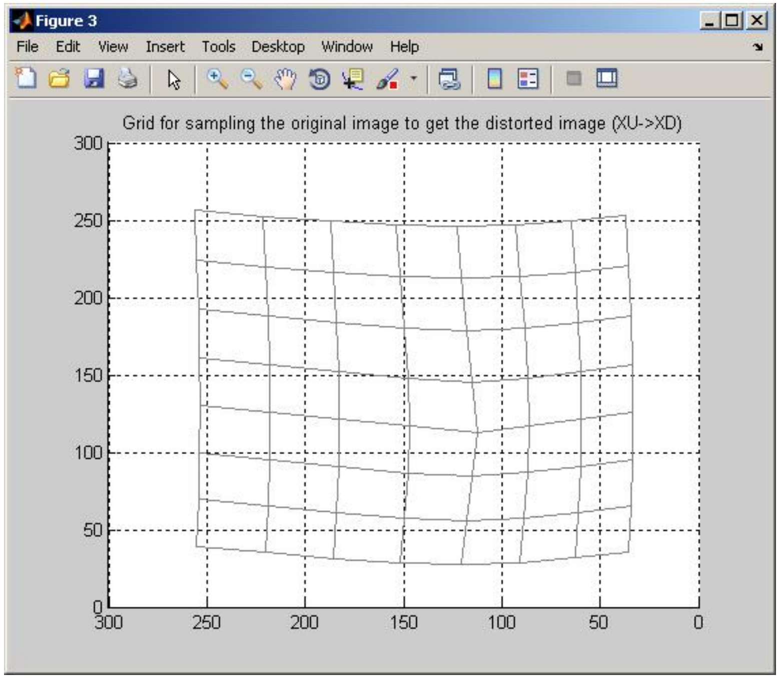
Task: Fill Out Feature Points



```
for r = 1:ny
   for c = 1:nx
         XU2(r,c) = XU(?, ?); % you may use floor()
         YU2(r,c) = …
         XD2(r,c) = …
         YD2(r,c) = …
   end
end
```

Visualize Coarse Grid



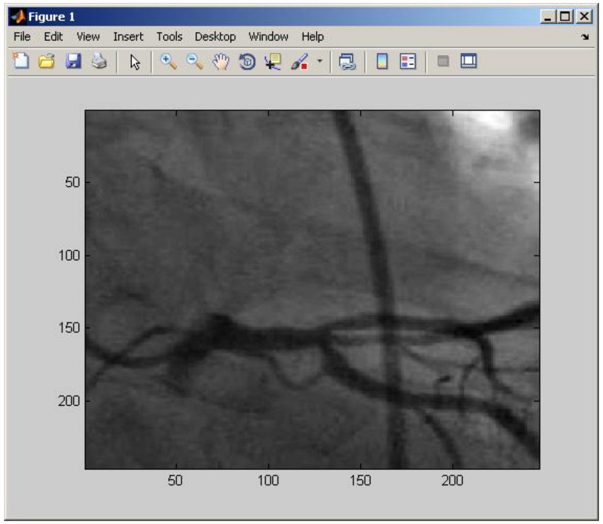


```
meshc(XU2, YU2, B)
```

```
meshc(XD2, YD2, B)
```

Image Undistortion

Task: Compute Distorted Points



Artificial Distortion

Undistortion

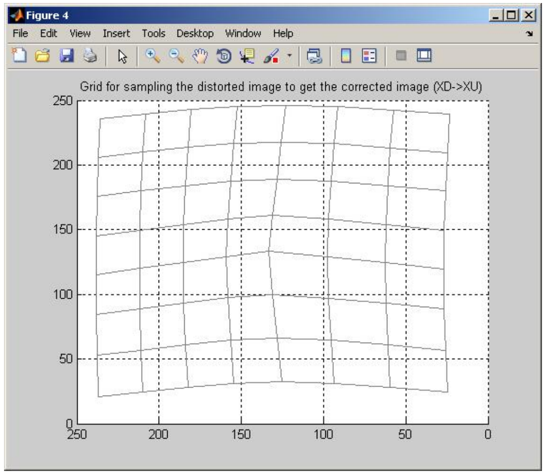
We now “distort” the distorted image to get the original image

Be aware of the fact, that the artificial deformation takes place from the distorted to undistorted. For creation, we used

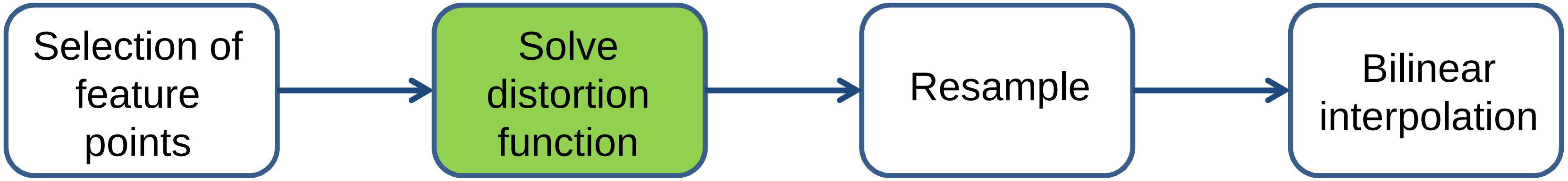
distorted = undistorted + deformation

```
XD2 = XU2 + (XU2 - XD2);
YD2 = YU2 + (YU2 - YD2);

meshc(XD2,YD2,B);
```



Distortion Function



Selection of feature points → Solve distortion function → Resample → Bilinear interpolation

$$\underset{\overset{\downarrow}{XD2}}{x_r} = \sum_{i=0}^{d}\sum_{j=0}^{d-i} u_{ij}\, \underset{\overset{\downarrow}{YU2}}{y_r'^{j}}\, \underset{\overset{\downarrow}{XU2}}{x_r'^{i}} \qquad \underset{\overset{\downarrow}{YD2}}{y_r} = \sum_{i=0}^{d}\sum_{j=0}^{d-i} v_{ij}\, \underset{\overset{\downarrow}{YU2}}{y_r'^{j}}\, \underset{\overset{\downarrow}{XU2}}{x_r'^{i}}$$

What we try to solve is u_i,j and v_i,j

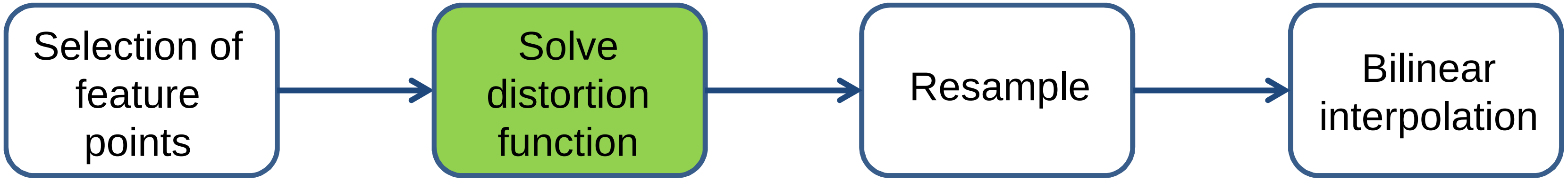
Distortion function can be rewritten in a matrix form (x_r for instance)

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \enclose{circle}{\boldsymbol{A}} \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,0} \end{pmatrix} \Rightarrow \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,d} \end{pmatrix} = \boldsymbol{A}^{\dagger} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Task: Create the measurement matrix A containing the polynomials.

Image Undistortion

Measurement Matrix A



Selection of feature points

Solve distortion function

Resample

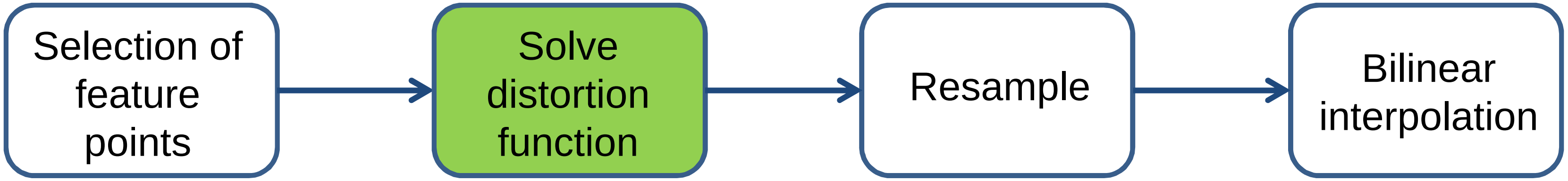
Bilinear interpolation

$$x_r = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} y_r'^{j} x_r'^{i} \quad \Rightarrow \quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,0} \end{pmatrix}$$

$$A = \overbrace{\begin{pmatrix} x_1'^0 y_1'^0 & x_1'^0 y_1'^1 & \dots & x_1'^0 y_1'^d & x_1'^1 y_1'^0 & \dots & x_1'^1 y_1'^{d-1} & x_1'^2 y_1'^0 & \dots & x_1'^d y_1'^0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ x_r'^0 y_r'^0 & x_r'^0 y_r'^1 & \dots & x_r'^0 y_r'^d & x_r'^1 y_r'^0 & \dots & x_r'^1 y_r'^{d-1} & x_r'^2 y_r'^0 & \dots & x_r'^d y_r'^0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ x_n'^0 y_n'^0 & x_n'^0 y_n'^1 & \dots & x_n'^0 y_n'^d & x_n'^1 y_n'^0 & \dots & x_n'^1 y_n'^{d-1} & x_n'^2 y_n'^0 & \dots & x_n'^d y_n'^0 \end{pmatrix}}^{\texttt{NumKoeff}} \left.\begin{matrix} \\ \\ \\ \\ \\ \end{matrix}\right\} \texttt{NumCorresp}$$

Image Undistortion

Measurement Matrix A



Selection of feature points → Solve distortion function → Resample → Bilinear interpolation

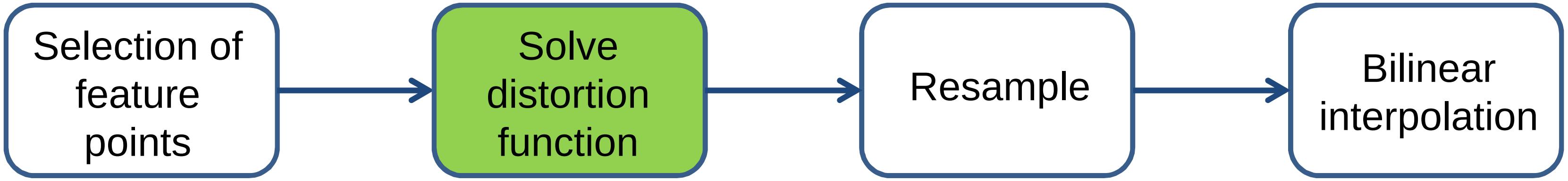
$$x_r = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} y_r'^{j} x_r'^{i} \quad \Longrightarrow \quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,0} \end{pmatrix}$$

Number of coefficients :

i = 0; j = 0, 1, 2, …, d	d+1
	+
i = 1; j = 0, 1, 2, …, d-1	d
	+
i = 2; j = 0, 1, 2, …, d-2	d-1
	+
…	…
	+
i = d-1; j = 0, 1	2
	+
i = d; j = 0	1

Image Undistortion

Measurement Matrix A


Selection of feature points

Solve distortion function

Resample

Bilinear interpolation

$$x_r = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} y_r'^j x_r'^i$$

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,0} \end{pmatrix}$$

Number of coefficients :

i = 0; j = 0, 1, 2, …, d	d+1
	+
i = 1; j = 0, 1, 2, …, d-1	d
	+
i = 2; j = 0, 1, 2, …, d-2	d-1
	+
…	…
	+
i = d-1; j = 0, 1	2
	+
i = d; j = 0	1

$$\Rightarrow n_{coeff} = \sum_{k=1}^{d+1} k = \frac{(d+1)(d+2)}{2}$$

Image Undistortion

Measurement Matrix A

Selection of feature points → **Solve distortion function** → Resample → Bilinear interpolation

```
NumKoeff = (d + 2) * (d + 1) / 2;
NumCoresp = size(XD2, 1) * size(YD2, 2);
```

$$A = \left.\overbrace{\begin{pmatrix} x_1'^0 y_1'^0 & x_1'^0 y_1'^1 & \dots & x_1'^0 y_1'^d & x_1'^1 y_1'^0 & \dots & x_1'^1 y_1'^{d-1} & x_1'^2 y_1'^0 & \dots & x_1'^d y_1'^0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ x_r'^0 y_r'^0 & x_r'^0 y_r'^1 & \dots & x_r'^0 y_r'^d & x_r'^1 y_r'^0 & \dots & x_r'^1 y_r'^{d-1} & x_r'^2 y_r'^0 & \dots & x_r'^d y_r'^0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ x_n'^0 y_n'^0 & x_n'^0 y_n'^1 & \dots & x_n'^0 y_n'^d & x_n'^1 y_n'^0 & \dots & x_n'^1 y_n'^{d-1} & x_n'^2 y_n'^0 & \dots & x_n'^d y_n'^0 \end{pmatrix}}^{\mathtt{NumKoeff}} \right\} \mathtt{NumCorresp}$$

Task: Fill Out Measurement Matrix

$$x_r = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} y_r'^{j} x_r'^{i} \quad \Longrightarrow \quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,0} \end{pmatrix}$$

`XU2vec(r)` = x'_r : Undistorted grid points’ x-coordinates

`YU2vec(r)` = y'_r : Undistorted grid points’ y-coordinates

```
for r = 1:NumCorresp
 c = 1;
 for i = 0:d
    for j = 0:(d-i)
        A(r,c) = ...;
        c = c + 1;
    end
  end
end
```

Image Undistortion

Task: Compute Pseudo-inverse of A

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,0} \end{pmatrix} \Rightarrow \begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,d} \end{pmatrix} = \boldsymbol{A}^{\dagger} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

```
[U, S, V] = svd(A);
…
Si = S';
epsilon = 1e-5;

for i = 1:size(S,2)
        if(S(i,i) < epsilon)
            Si(i,i) = …;
    else
            Si(i,i) = …;
 end
end
Apseudoinv  = …
```

$$A = U\Sigma V^T$$

Set singular values lower than 10E-5 to zero for a better conditioned equation system.

$$A^{\dagger} = V\Sigma^{-1}U^T$$

Task: Compute Coefficients

Compute the distortion coefficients u_i,j, v_i,j

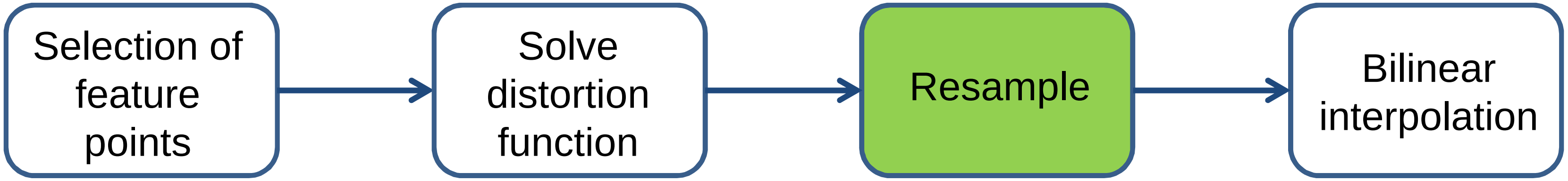
`XD2vec(r)` = x_r : Distorted grid points’ x-coordinates

`YD2vec(r)` = y_r : Distorted grid points’ y-coordinates

```
Uvec = …; % u_i,j
Vvec = …; % v_i,j
```

$$\begin{pmatrix} u_{0,0} \\ u_{0,1} \\ \vdots \\ u_{d,d} \end{pmatrix} = \boldsymbol{A}^{\dagger} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Compute Fine Grid Points

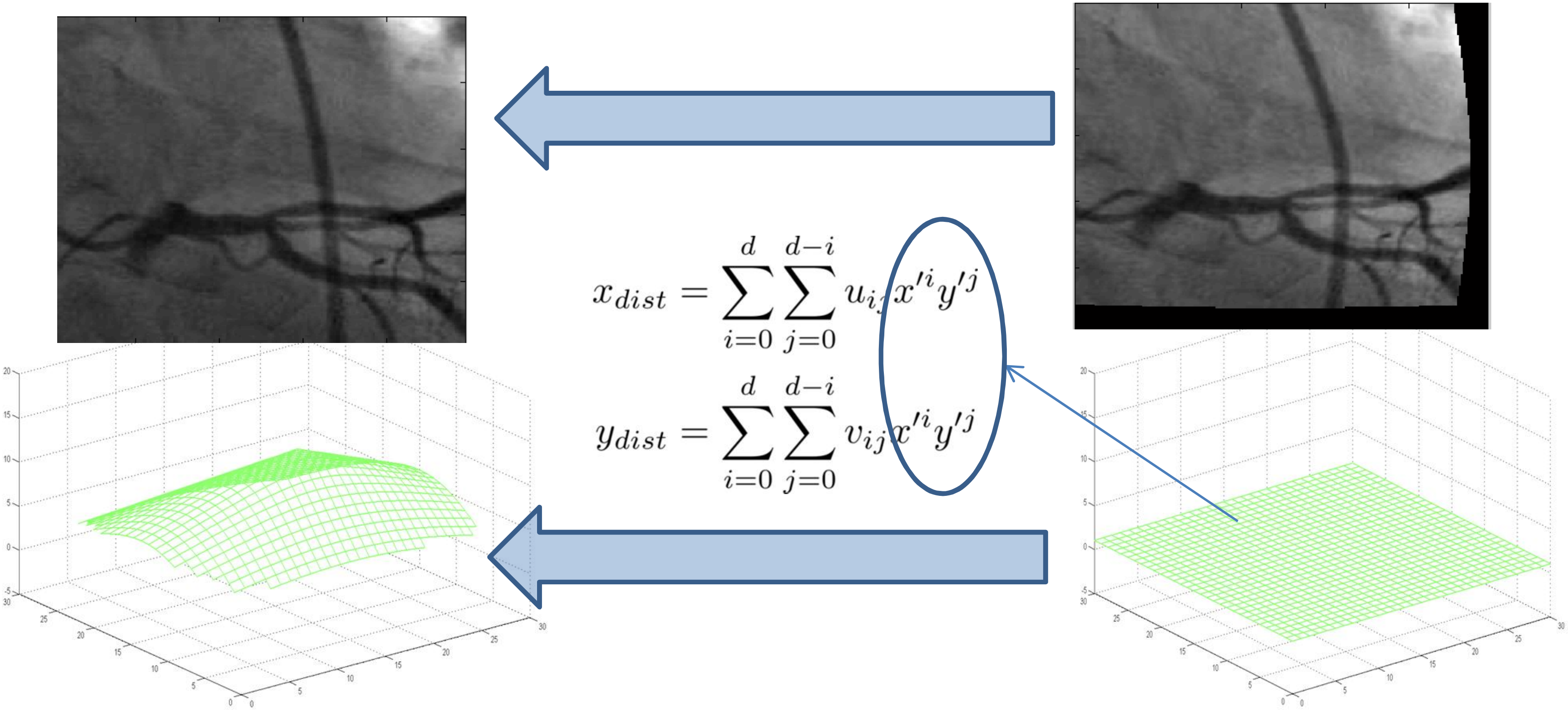


Selection of feature points

Solve distortion function

Resample

Bilinear interpolation

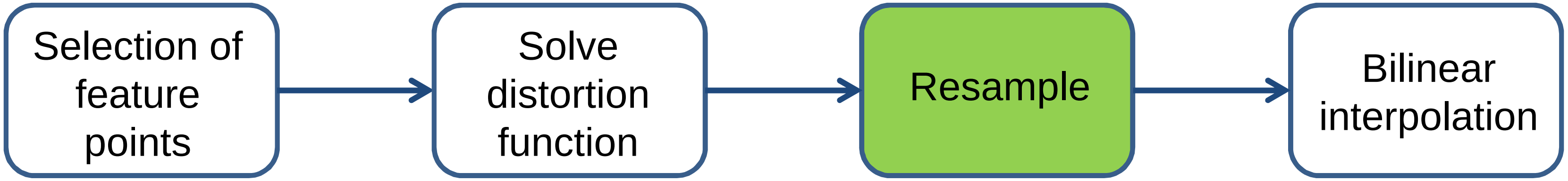


$$x_{dist} = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} x'^i y'^j$$

$$y_{dist} = \sum_{i=0}^{d} \sum_{j=0}^{d-i} v_{ij} x'^i y'^j$$

Task: Compute the grid points which are used to sample the distorted image to get the undistorted image.

Compute Fine Grid Points



Selection of feature points

Solve distortion function

Resample

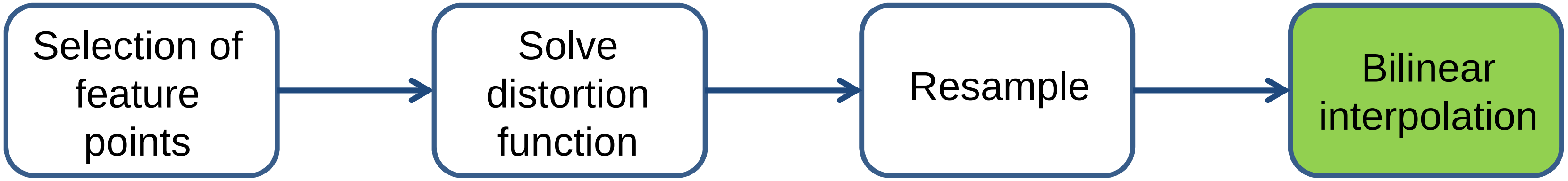
Bilinear interpolation

```
for y = 1:sy
 for x = 1:sx
     c = 1;
     for i = 0:d
             for j = 0:(d-i)
             XDist(y,x) = …;
             YDist(y,x) = …;
             c = c + 1;
             end
     end
     end
end
```

$$x_{dist} = \sum_{i=0}^{d} \sum_{j=0}^{d-i} u_{ij} {x'}^i {y'}^j$$

$$y_{dist} = \sum_{i=0}^{d} \sum_{j=0}^{d-i} v_{ij} {x'}^i {y'}^j$$

2D Interpolation

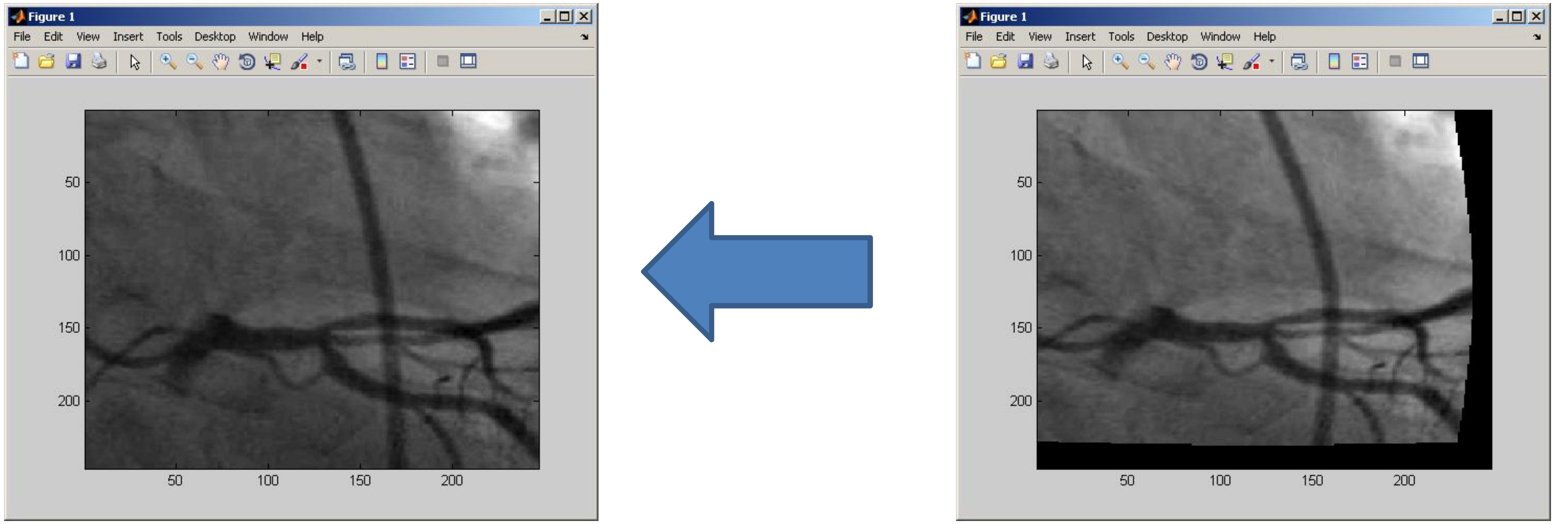


Selection of feature points

Solve distortion function

Resample

Bilinear interpolation



Create an corrected image `undist` at the current grid positions `X,Y` where the intensities are interpolated at the positions `XDist, YDist` in `Idist`:

```
undist = …;
undist(isnan(undist)) = 0;
```

Image Undistortion

6. Scaling of Input Data

Think about it! Do you have a good feeling in doing this?

- Use a polynomial of total degree 5 to undistort images.
- Input images are 1024 x 1024–image.
- The x and y coordinates are represented in pixels, i.e.
- $x, y \in$ 1, 2, … 1024
- The monomials range from 1 to $1024^5 = 1125899906842624$
- The result has to be between 0 and 1023!!!

Image Undistortion

6. Scaling of Input Data

The Gramian matrix can be used to test for linear independence of functions.

Any decrease of the condition number will be useful, even if it is not a global optimum!

Method to compute a proper scaling:

- Select constants k and l
- Scale all data points (x_i, y_i) to (kx_i, ly_i)
- Rewrite (9) and compute new A
- Compute condition number (A^TA)
- Minimize with respect to k and l, e.g. by gradient descent
- Finally, recover the original coefficients $u_{i;j}$, $v_{i;j}$ and invert the scaling process