Computer Vision
and Geometry Lab

Computer Vision

Exercise Session 1

Organization

- Teaching assistant
  - Bastien Jacquet
    - CAB G81.2
    - bastien.jacquet@inf.ethz.ch
  - Federico Camposeco
    - CNB D102.2
    - fede@inf.ethz.ch
- Lecture webpage
  - http://www.cvg.ethz.ch/teaching/compvis/index.php

Organization

- Assignments will be part of the final grade
  - Assignment every week (or every two weeks the second part)
  - 1 or 2 weeks time to solve the assignment
  - Hand-in Thursday 13:00 (sharp!)
  - Bonus points can be used to compensate for missing points
- MATLAB Download (www.stud-ides.ethz.ch)

Literature

- Computer Vision: Algorithms and Application
  by Richard Szeliski, available online (http://szeliski.org/Book/)
- Multiple View Geometry
  by Richard Hartley and Andrew Zisserman
- Course Notes
  - http://cvg.ethz.ch/teaching/compvis/tutorial.pdf

Camera Calibration

- Intrinsic parameters
  - **K**
  - Radial distortion coefficients

2D points → $\mathbf{x}$

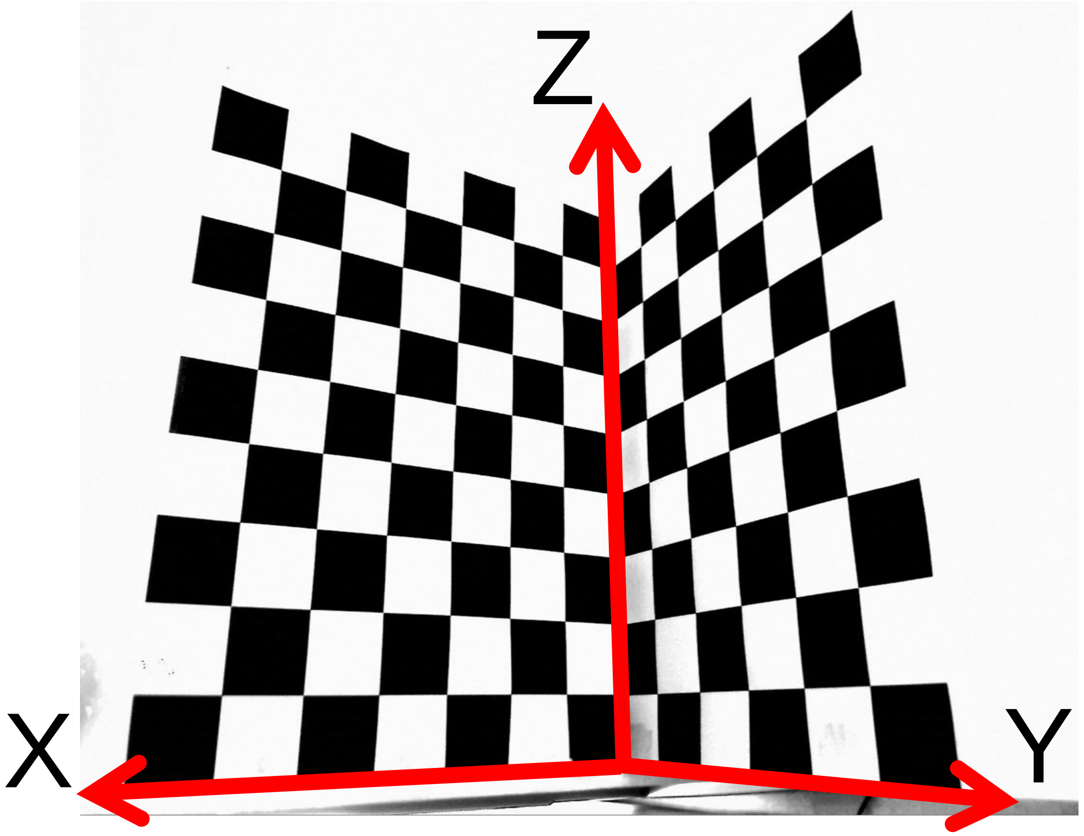
3D points → $\mathbf{X}$

$$\mathbf{x} \propto \mathbf{P}\mathbf{X}$$

$$\mathbf{x} \propto \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\mathbf{X}$$

Camera Calibration

- Use your own camera
- Build your own calibration object
  - Print checkerboard patterns
  - Stich to two orthogonal planes



Camera Calibration

- 4 Tasks:
  - Data normalization
  - Direct Linear Transform (DLT)
  - Gold Standard algorithm
  - Bouguet‘s Calibration Toolbox
- Use the same settings for all tasks!
- Good reference:
  Multiple View Geometry in computer vision
  (Richard Hartley & Andrew Zisserman)

Data normalization

- Shift the centroid of the points to the origin
- Scale the points so that average distance to the origin is $\sqrt{3}$ and $\sqrt{2}$, respectively.
- Determine $\hat{\mathbf{P}}$ using normalized points.
- Determine $\mathbf{P} = \mathbf{T}^{-1}\hat{\mathbf{P}}\mathbf{U}$

$$\mathbf{T} = \begin{bmatrix} s_{2D} & & c_x \\ & s_{2D} & c_y \\ & & 1 \end{bmatrix}^{-1}$$

$$\mathbf{U} = \begin{bmatrix} s_{3D} & & & c_x \\ & s_{3D} & & c_y \\ & & s_{3D} & c_z \\ & & & 1 \end{bmatrix}^{-1}$$

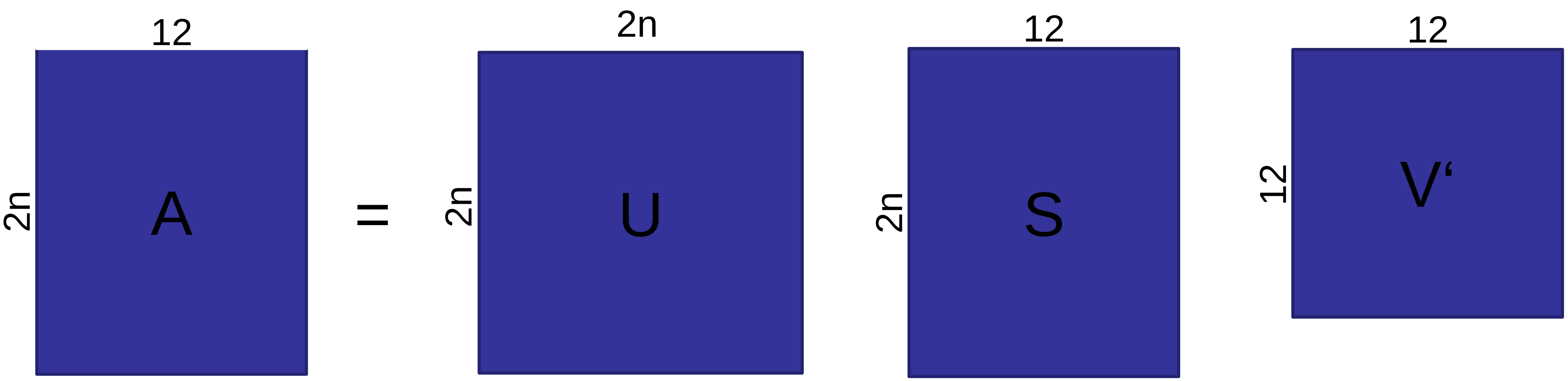
Direct Linear Transform (DLT)

$$\mathbf{AP} = \begin{bmatrix} w_i X_i^T & 0^T & -x_i X_i^T \\ 0^T & -w_i X_i^T & y_i X_i^T \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = 0$$

$$= \begin{bmatrix} X_{ix} & X_{iy} & X_{iz} & 1 & 0 & 0 & 0 & 0 & -x_i X_{ix} & -x_i X_{iy} & -x_i X_{iz} & -x_i \\ 0 & 0 & 0 & 0 & -X_{ix} & -X_{iy} & -X_{iz} & -1 & y_i X_{ix} & y_i X_{iy} & y_i X_{iz} & y_i \end{bmatrix} \begin{pmatrix} P_{1,1} \\ P_{1,2} \\ \vdots \\ P_{3,3} \\ P_{3,4} \end{pmatrix}$$

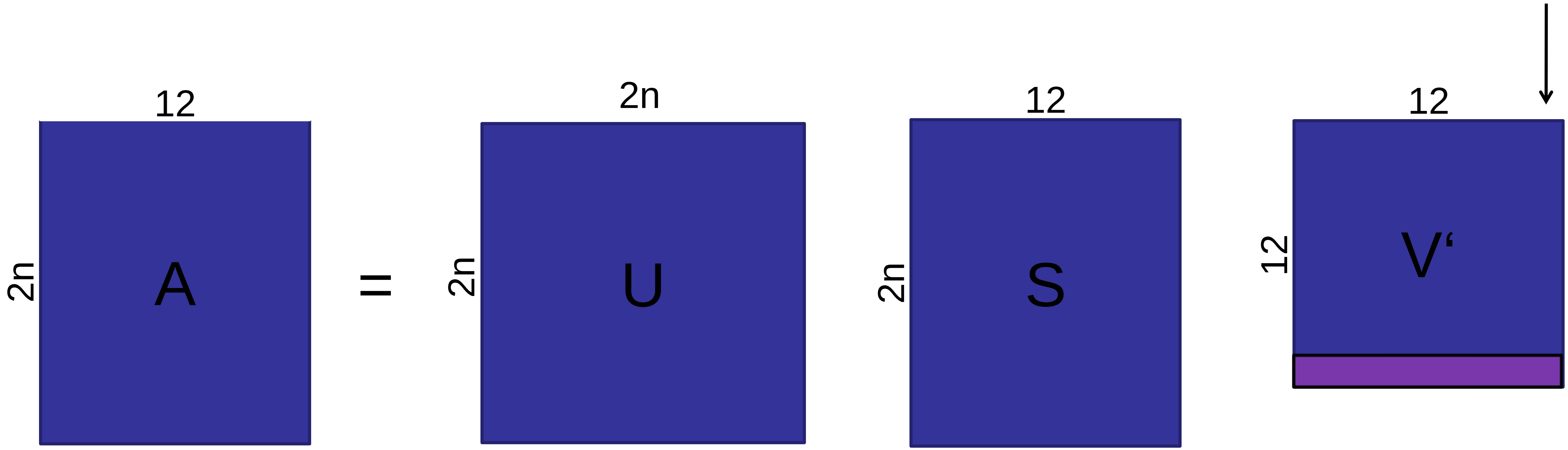
Direct Linear Transform (DLT)

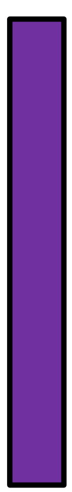
- Singular Value Decomposition



Direct Linear Transform (DLT)

- Singular Value Decomposition





$$= \begin{pmatrix} P_{1,1} \\ P_{1,2} \\ \vdots \\ P_{3,3} \\ P_{3,4} \end{pmatrix} \longrightarrow \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \end{bmatrix}$$

Camera Matrix Decomposition (K and R)

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right] = \left[\mathbf{KR} \mid -\mathbf{KRC}\right]$$

- **K** is upper triangular
- **R** is orthonormal
- **QR** decomposition **A** = **QR**
  - **Q** is orthogonal
  - **R** is upper triangular

Camera Matrix Decomposition (K and R)

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right] = \left[\mathbf{KR} \mid -\mathbf{KRC}\right]$$

$$\mathbf{M} = \mathbf{KR}$$

$$\mathbf{M}^{-1} = \mathbf{R}^{-1}\mathbf{K}^{-1}$$

- Run **QR** decomposition on the inverse of the left 3x3 part of **P**
- Invert both result matrices to get **K** and **R**

Camera Matrix Decomposition (C)

- The camera center is the point for which

$$\mathbf{PC} = 0$$

- This is the right null vector of **P** (→ SVD)

Gold Standard Algorithm

- Normalize data
- Run DLT to get initial values
- Compute optimal $\hat{\mathbf{P}}$ by minimizing the sum of squared reprojection errors

$$\min_{\hat{\mathbf{P}}} \sum_{i=1}^{N} d(\hat{\mathbf{x}}_i, \hat{\mathbf{P}}\hat{\mathbf{X}}_i)^2$$

- Denormalize $\hat{\mathbf{P}}$

Minimization in MATLAB

- `Fminsearch(...)`
  - See code framework

- `Lsqnonlin(...)`
  - nonlinear least-squares

- Vectorize your parameters

Bouguet‘s Calibration Toolbox

- Download and install the toolbox:
  (http://www.vision.caltech.edu/bouguetj/calib_doc/index.html)
- Go through the tutorial and learn how to calibrate a camera with that toolbox
- Print your own calibration pattern (available on the website)
- Use the toolbox for calibration and compare the result with the results of your own calibration algorithm

Hand-in

- Source code
- Matlab .mat file with hand-clicked 3D-2D correspondences
- Image used for calibration
- Visualize hand-clicked points and reprojected 3D points
- Discuss and compare values of calibration obtained for all methods
- Discuss average reprojection error of all methods.

Some hints

- Work with normalized homogeneous coordinates always. Camera calibration **K** should respect this convention.
- Check that the obtained orientation **R** correspond to the expected world value, otherwise **K** will have negative values in its diagonal.
- When reprojecting points use average of reprojection error for comparison and remember to normalize reprojected coordinates to w = 1.
- Remember to use the same camera with the same settings for all tasks!

Hand-in

- Example reprojection of the the 3D points