3D COMPUTER VISION

TWO-VIEW SPARSE 3D RECONSTRUCTION AND DEPTH-BASED SEGMENTATION

E5ADSB

Janus Bo Andersen

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN AARHUS UNIVERSITET

GOALS OF THIS PRESENTATION

Present project: ~10 minutes

Discussion of project and curriculum: ~5-10 minutes

Goals of presentation:

- Summarise **project and key results**
- Summarise key **conclusions**, future work with emphasis on **interesting applications**
- Describe and discuss **pipeline**, with emphasis on steps implemented by author
- **Questions**

https://aarhusuniversity.zoom.us/j/63067822174

PROJECT SUMMARY

Situation / background

- 3D information is lost during imaging
  - → recover some of this lost detail to segment an image based on objects' distances to camera

Problems

- Tracking: Mark locations that represent same world point in sets of two views of same scene
- Recovery:
  - Estimate relative camera pose (up to scale)
  - Recover world points (up to scale)
- Application: Use distance vectors in world frame to segment image

Key result

- Developed and implemented pipeline to solve problems
- Obtained depth-segmented image
- Report that shows "process" and leads up to further work.

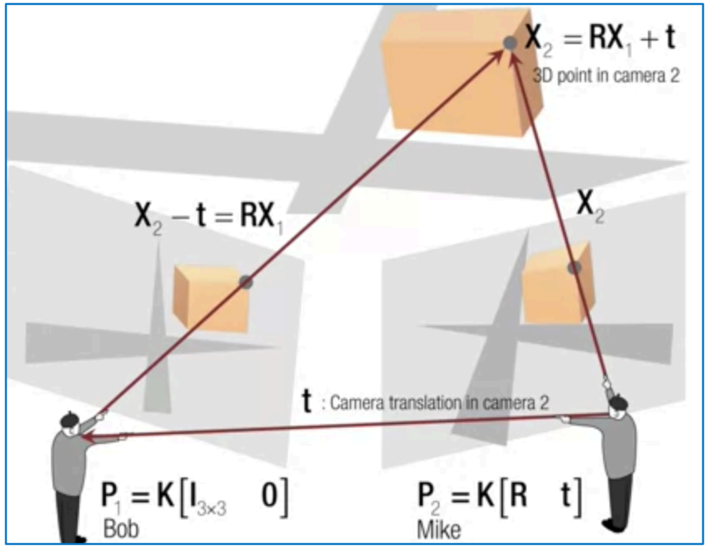
ILLUSTRATION OF PROBLEM

Core problem is illustrated in figure

- Relating points in pairs of images to determine [R, t] and triangulate.

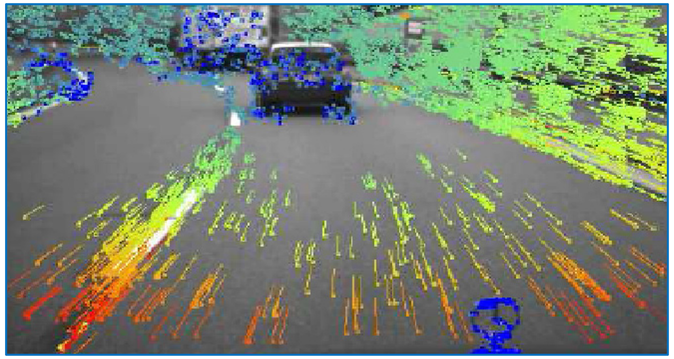
Baseline size guides approach

- <u>Large</u> baseline
  - -> Case as in this project
  - **Feature tracking** + Multiple-view geometry / Structure from motion (SfM)
- <u>Small</u> baseline
  - -> Video
  - Optical flow + Stereo disparity



Geometry of two views

Large baselines: Feature tracking



Small baselines: Optical flow

KEY CONCLUSIONS AND FUTURE DIRECTION

Key conclusions

System works as intended

- According to requirements.
- Many ideas for improvement and application.
- Acquisition step must be improved to reduce number of outliers.

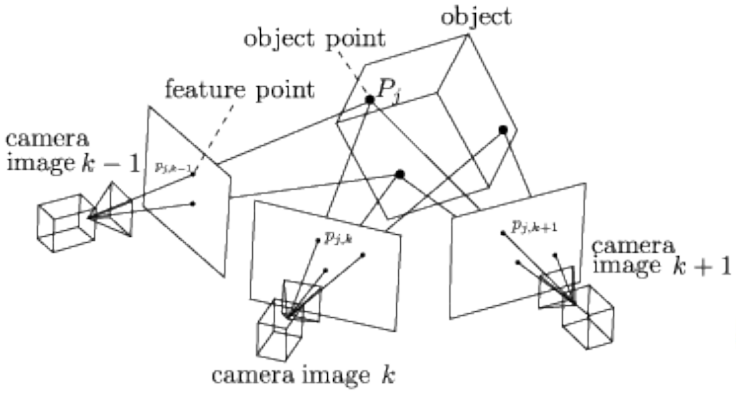
Feature tracking algorithm

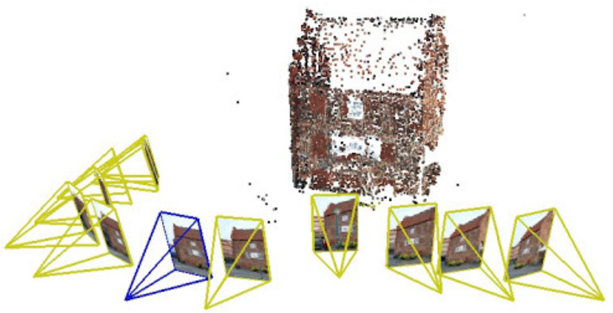
- Harris corners + BRIEF + brute-force Hamming Distance and Lowe's Ratio Test **work OK**.
  - Implemented Harris **corner detection algorithm is relatively slow**, not yet fit for real-time system.
  - The other algos are quite fast.
  - Tracking **not yet invariant to rotations**.

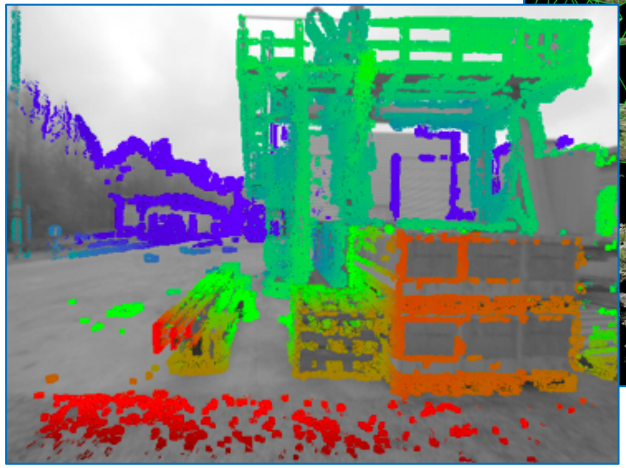
Depth-based segmentation from 3D vectors

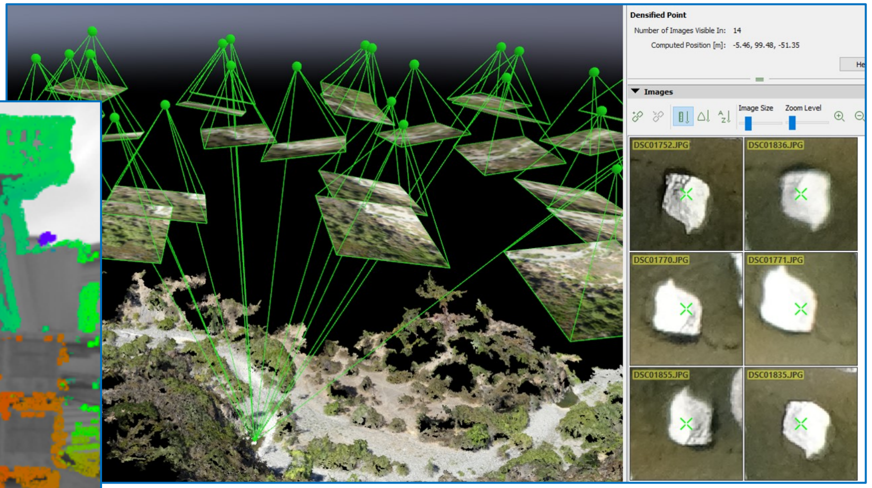
- Thresholding + morph. closing + labelling **works OK** to automatically segment image into regions.
  - However, somewhat **sensitive** to lighting conditions and relative contrasts as seen on e.g. the "dog food".
- Attaching 3D information from point cloud and majority voting **works OK**.

Direction from here is…









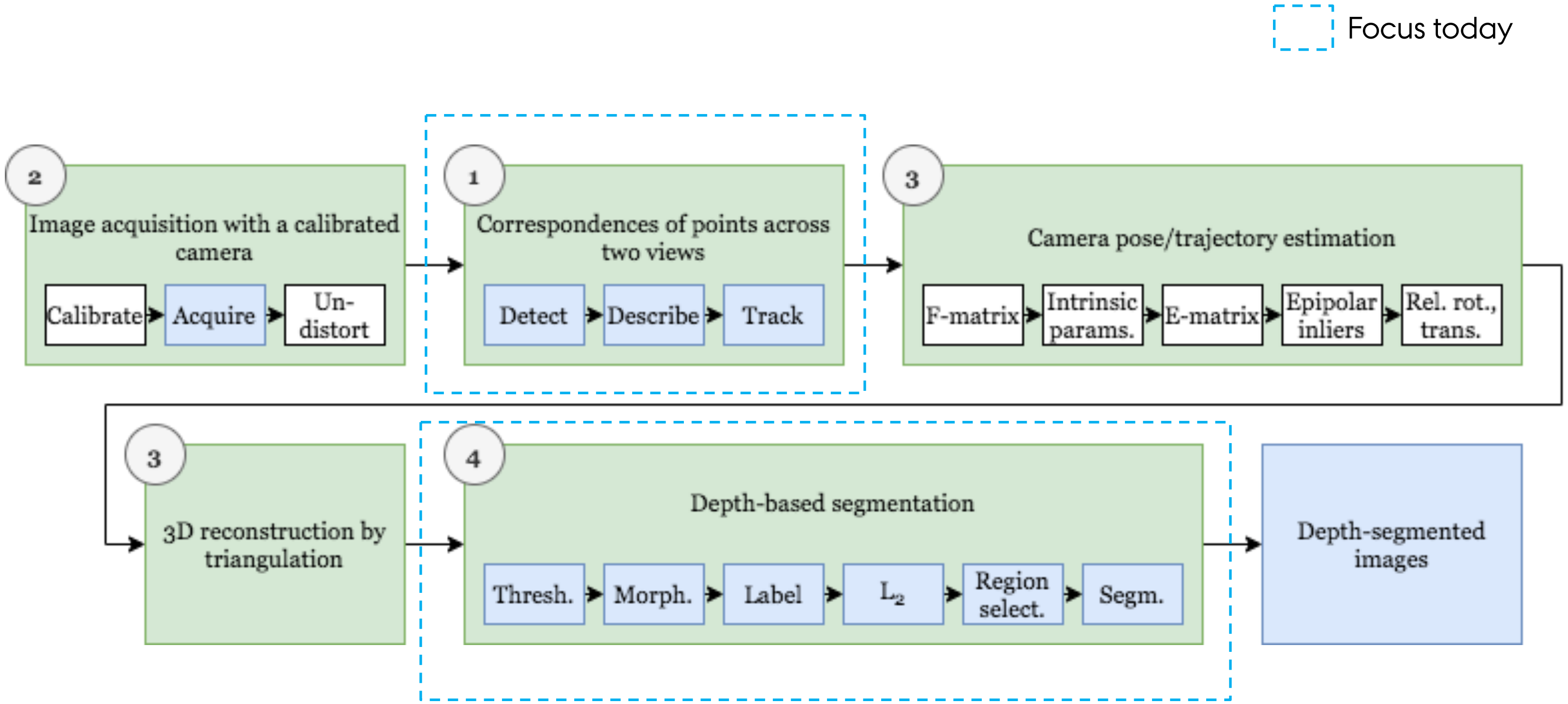
3D VISION PIPELINE

Pipeline steps developed in this project:

- **Correspondence** problem
  - Harris corner **detection**
  - BRIEF feature **description**
  - Two-view feature **tracking**
- Depth-based **segmentation** step

Researched theory and used Matlab algorithms for

- Camera calibration
- Relative pose estimation
- Triangulation, 3D reconstruction



CORRESPONDENCE STEP

Generally relevant in computer vision, not just for 3D vision

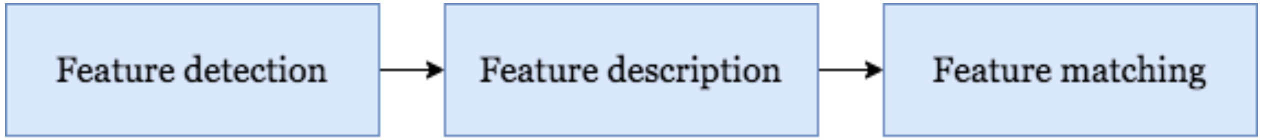
- Three steps

Algo choices for this project

- Harris corners very common
- Non-max suppression was necessary
- “Simpler” description/matching algorithms that can be implemented on real-time systems
- Harris detector not yet fast enough for that...

Next extension is ORB features

- Oriented FAST and Rotated BRIEF:
  - FAST: Computationally simpler corner detector
  - Rotated BRIEF: Rotate features along principal axes, to make feature descriptors rotationally invariant.



Locate many key points by a detection algorithm, e.g.

- Harris corners (corner structures).
- Differences of Gaussians (blobs, edges)
- SURF (Hessian, blobs)

Detection criteria, thresholding on e.g.

- Harris criterion (two large eigenvalues found from relative sizes).
- Shi-Tomasi (same but different weighting).

Non-max suppression to mark only 1 key point within region.

Close-to-unique description of each key point, "fingerprints". Build feature vectors, e.g.

- BRIEF (binary vector from binary pixel comparisons).
- SIFT (differences of gaussians at different scales).
- SURF (pyramid of different smoothing levels).
- ORB (rotated BRIEF features).

Recognition of key points from one view in another view by a distance metric, e.g.

- Hamming distance for BRIEF and ORB.
- Nearest neighbours in SIFT.

Brute-force matching compares all features to all others.

Outliers can be removed using majority consensus method and epipolar geometric constraint.

EXAMPLE OF CORRESPONDENCE

Feature **detection** -> requires sufficient local structure

- Blank wall / aperture problem
- Corners make good features
  - Localizable in two directions
  - Relatively invariant to perspective projection
  - Relatively invariant to lighting changes
- Significant spatial grad. in two dirs. -> corner
- Feature points visible in both views of the scene

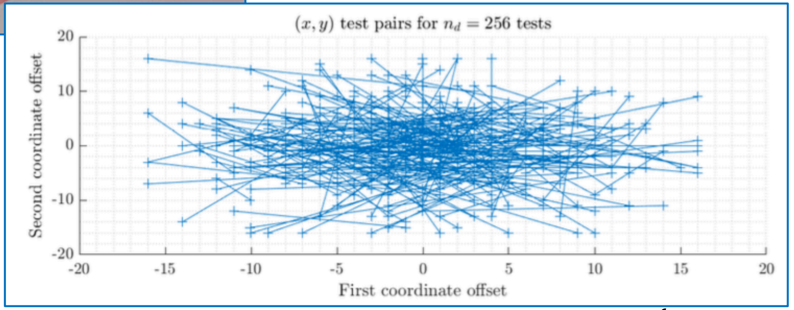
Feature **description** -> vector describing point n'hood

Feature **matching** -> with a binary vector, Hamming dist.

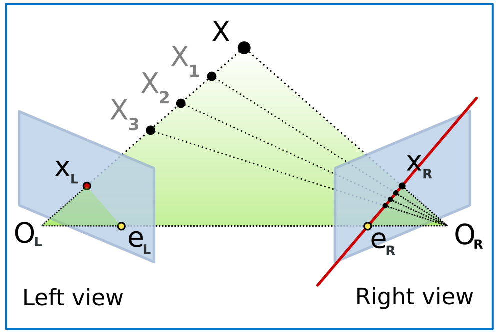
Some putative matches are later marked as epipolar outliers

- Likely due to imperfect camera calibration
- Acquisition step to be improved

Two-view correspondence



Describing feature n'hood by binary comparisons (grayscale)



Epipolar constraint – image point in one view must transforms to line in other view

SEGMENTATION STEP

3D vectors from triangulation after two-view corresp. and pose estim.:

- Far vs. near in point cloud is classified by mean L_2 distance

Image from view 1 pre-processed to get large connected regions:

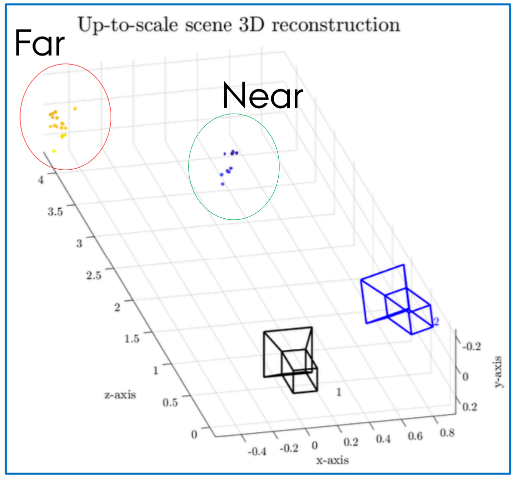
- Thresholding (Otsu '79)
- Morphological closing (disk)
- Fill holes
- Assign labels
- Attach the distance classifiers

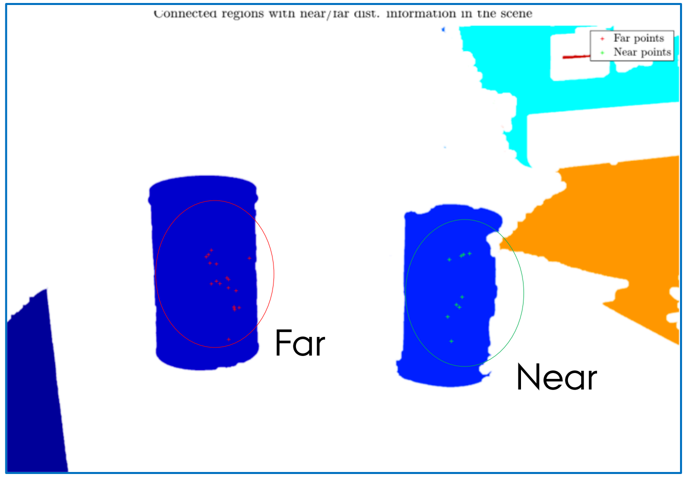
Selection and segmentation:

- E.g. which labelled region does the majority of "far" points lie in?

Advantages and drawbacks of this method:

- Advantage: Simple method, low complexity.
- Drawback: This step requires general scene knowledge, e.g. setup
  - the number of relevant classes to do dist. separation.
  - that the background is white to do threshold.





Pick out surface of far obj.

Region segmentation

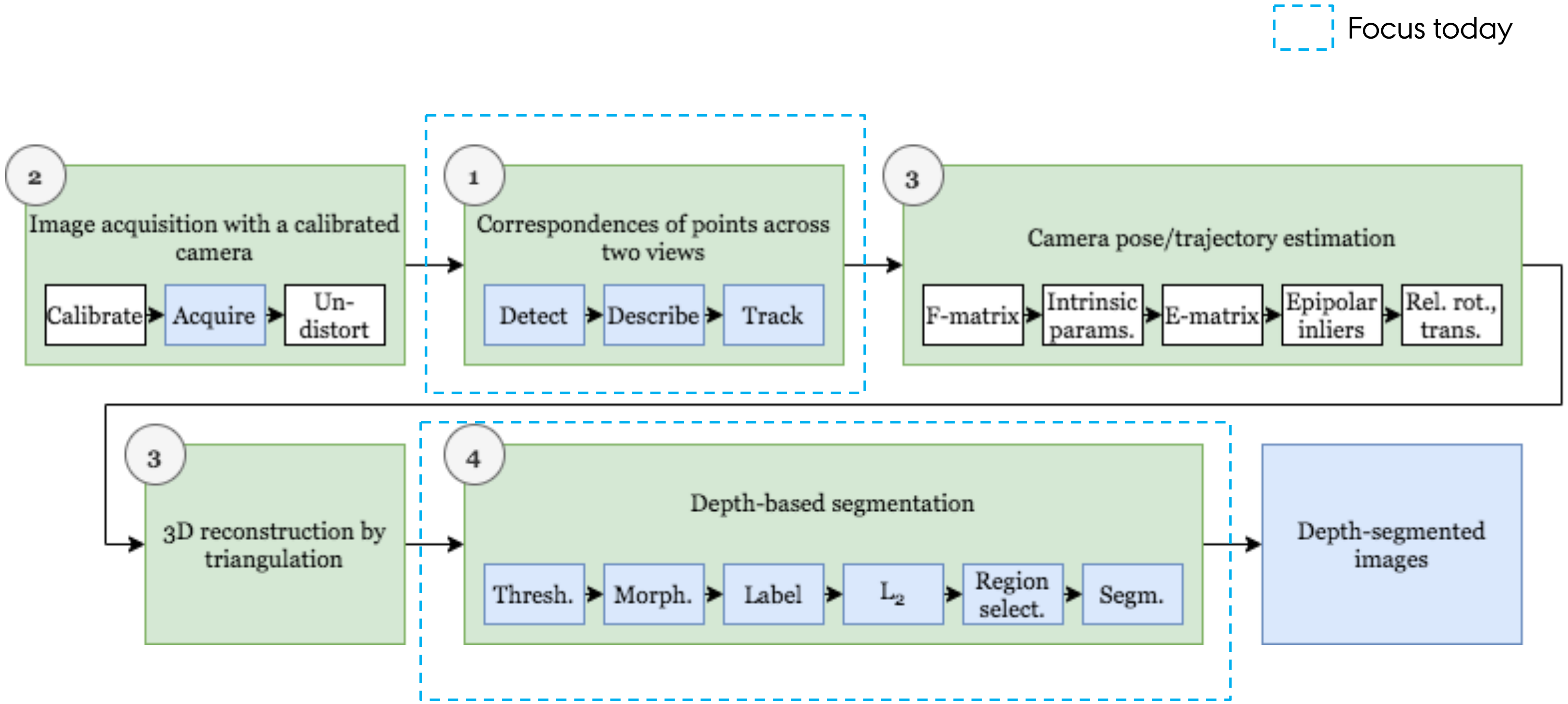
SUMMARY OF PRESENTATION

Discussed the 3D vision system

- Conclusion and future direction

Insight into implemented algorithms

- Improvement ideas
- Advantages and drawbacks



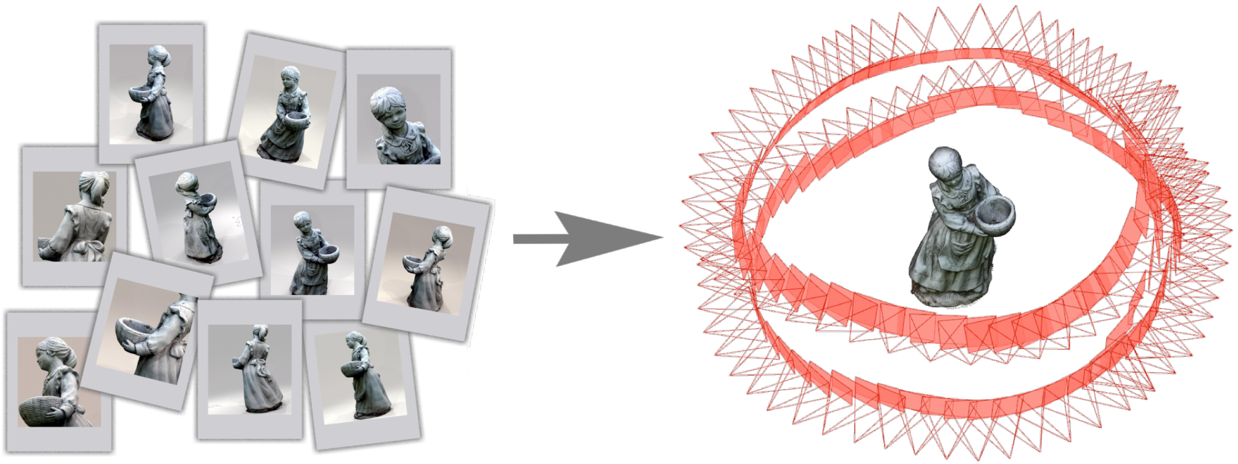
QUESTIONS

BACKUP SLIDES

FORMÅL OG ANVENDELSER

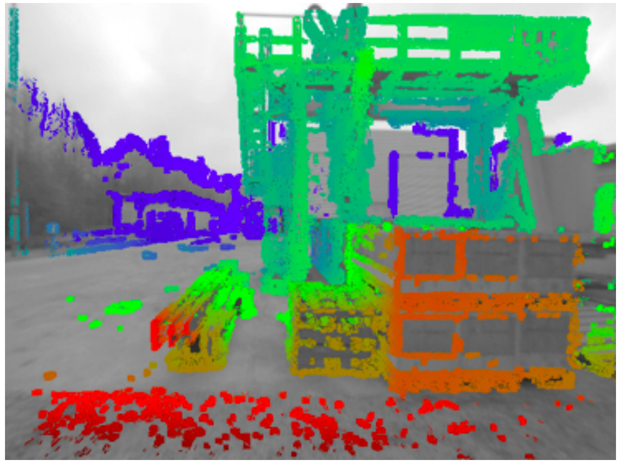
Udvind 3D-information fra 2D-billeder, fx dybde, struktur størrelser (metrologi)

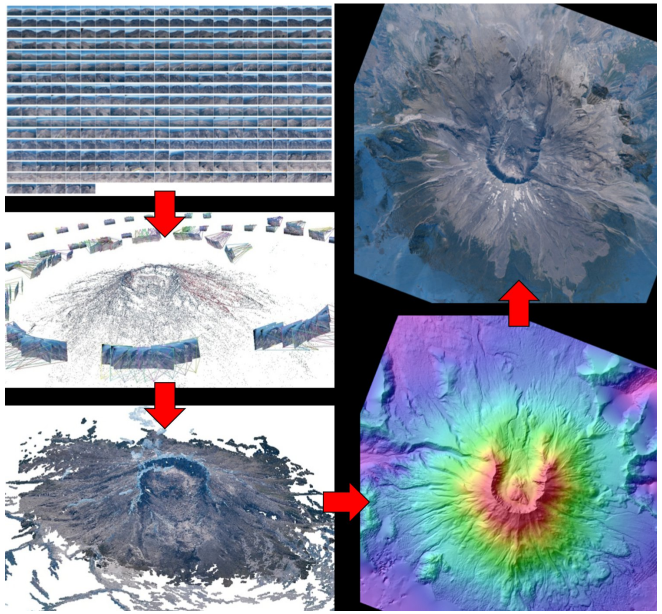
- "Single-view" geometri -> 1 billede / kamera
- "Multiple-view" geometri
  - Stereosyn (2-view) -> 2 billeder (øjne)
  - N-view -> Mange billeder / kamerakonfigurationer



Computer vision/perception - anvendelser?

- Afstandsbedømmelse, "hvor er jeg?" og SLAM (robotics, selvkørende biler, osv.)
- Visuelle effekter (augmented reality)
- Billedsegmentering med objektafstand
- 3D rekonstruktion af objekter
- Landmåling, geologiske undersøgelser
- Osv.





PROJEKTIV GEOMETRI: SINGLE-VIEW

PARALLELLE LINJER, RETTE VINKLER, FORSVINDINGSPUNKTER, OSV.

Oprindeligt billede

Vermeer's *Music Lesson*

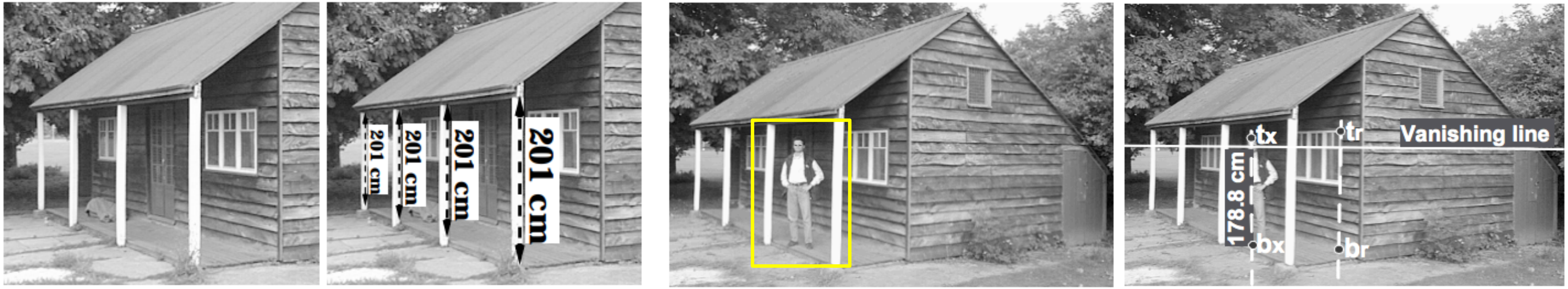
Rekonstrueret 3D-model

Rekonstrueret 3D-model

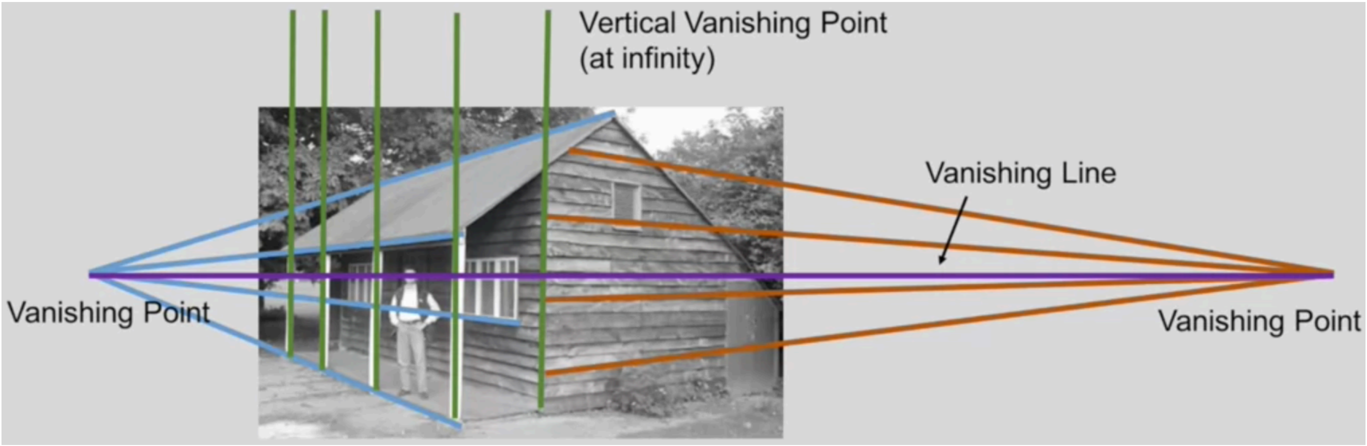
Criminisi et al. "Single view metrology", 1999

PROJEKTIV GEOMETRI: SINGLE-VIEW

METROLOGI: FX GIVET REFERENCERNE, HVOR HØJ ER MANDEN?



a b c d



STEREOSYN

Observation af punkt X giver (perspektiv-) projektionerne x_L og x_R.

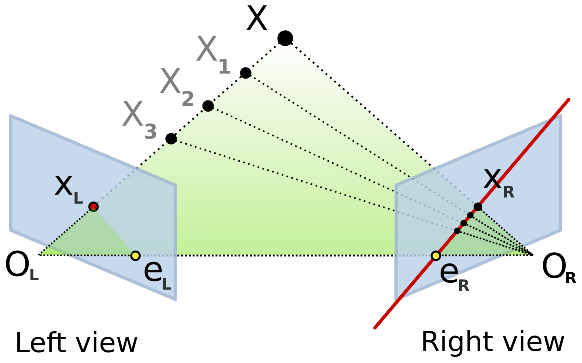
Hvis to kamerapositioner på forhånd er kendt / kalibreret:

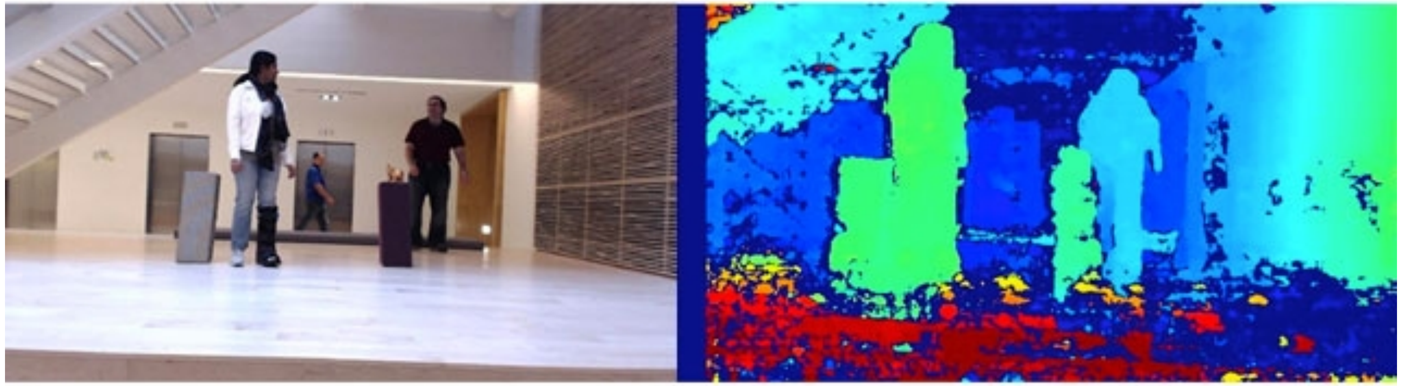
- "Triangulering" kan give position på X (relativt)

Eksempel, stereokamera benyttet til at estimere afstande i scenen

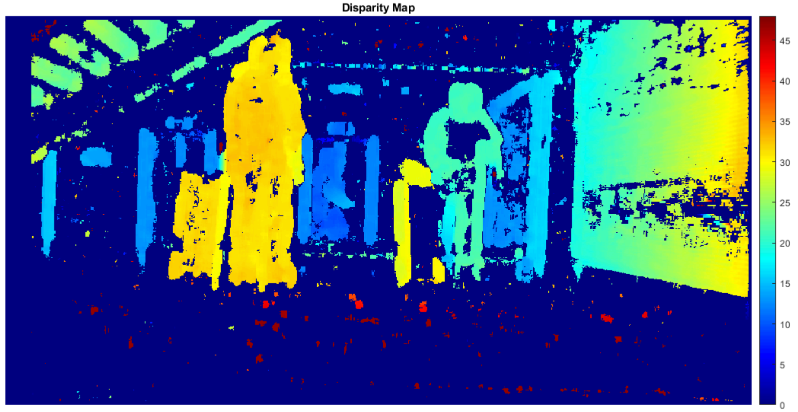
- Kan evt. segmentere billedet ud fra dybde

Red-Cyan composite view of the rectified stereo pair image





Disparity Map



ROTATION OG TRANSLATION AF KAMERA ESTIMERING AF STRUKTUR

Korrespondance mellem to billeder af samme scene, taget forskudt og med stort overlap

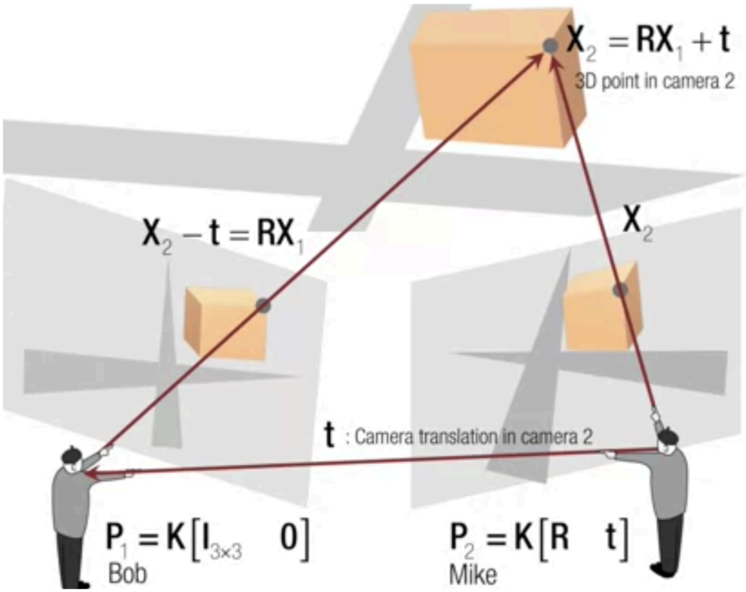
8 punkter (identificeret ud fra fx features)

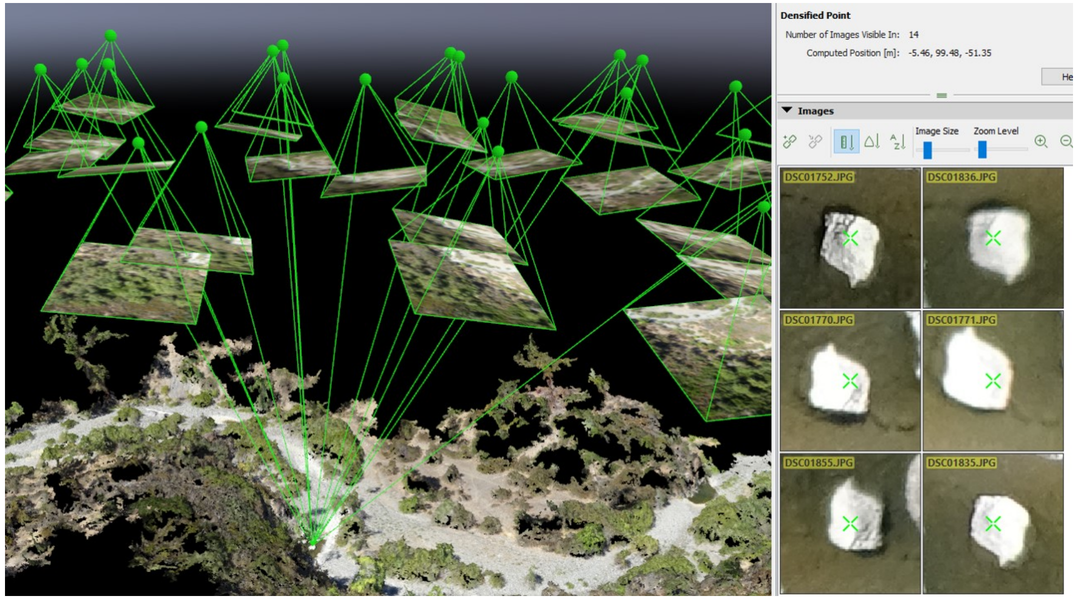
Benyt epipolære begrænsninger

- Geometrisk/matematisk relation mellem de to billeder (projektioner) og punkter i 3D
- Begrænsninger -> ligningssystem

Udregn relativ **bevægelse** (rotation og translation) af kamera/observatør -> giver projektionsmatricer

Beregn **struktur** ved triangulering og projektionsmatricer.



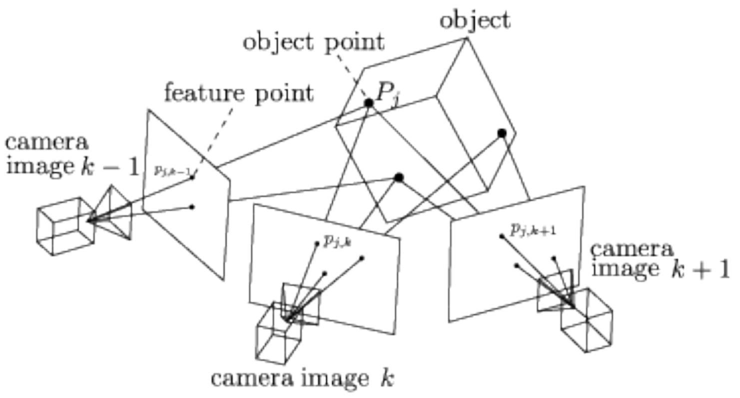


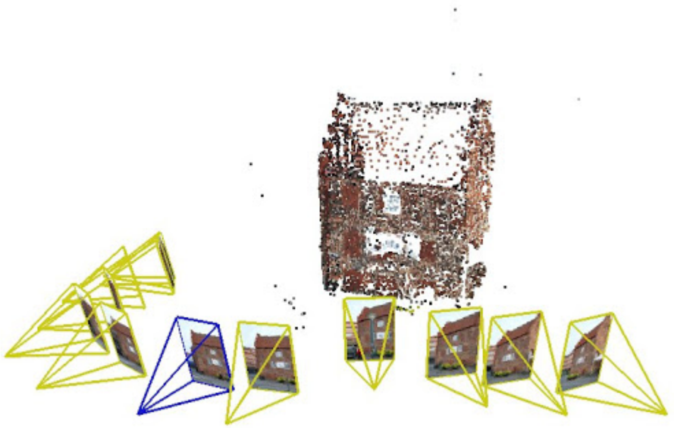
MULTIPLE-VIEW

Mange billeder fra mange positioner

Structure from motion (SfM)

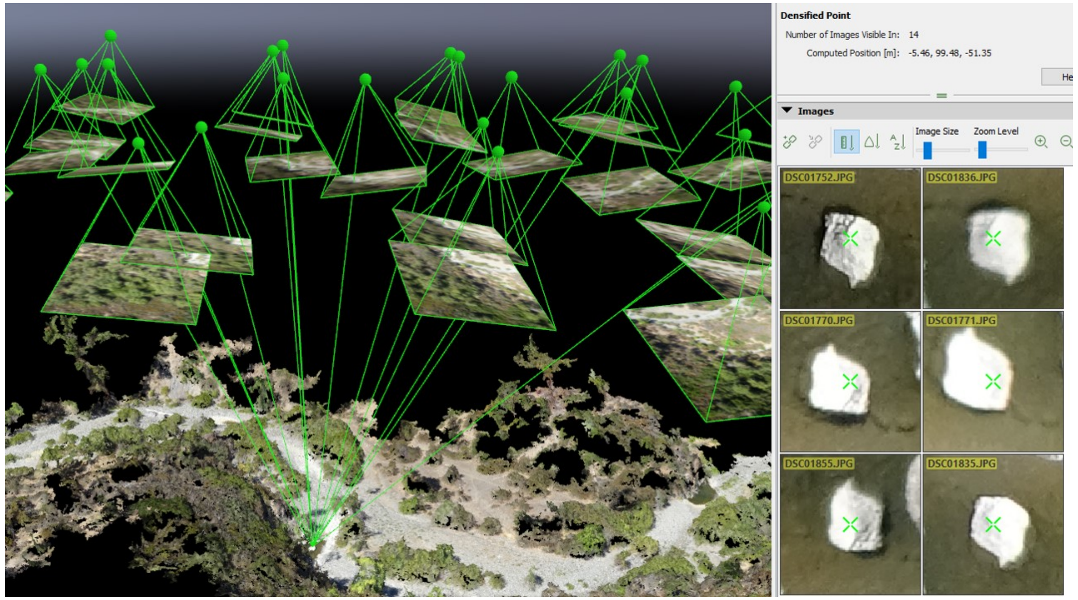
- ”Motion” er bevægelse af kameraet
- ”Structure” er den ting, der observeres og rekonstrueres i 3D





Proces:

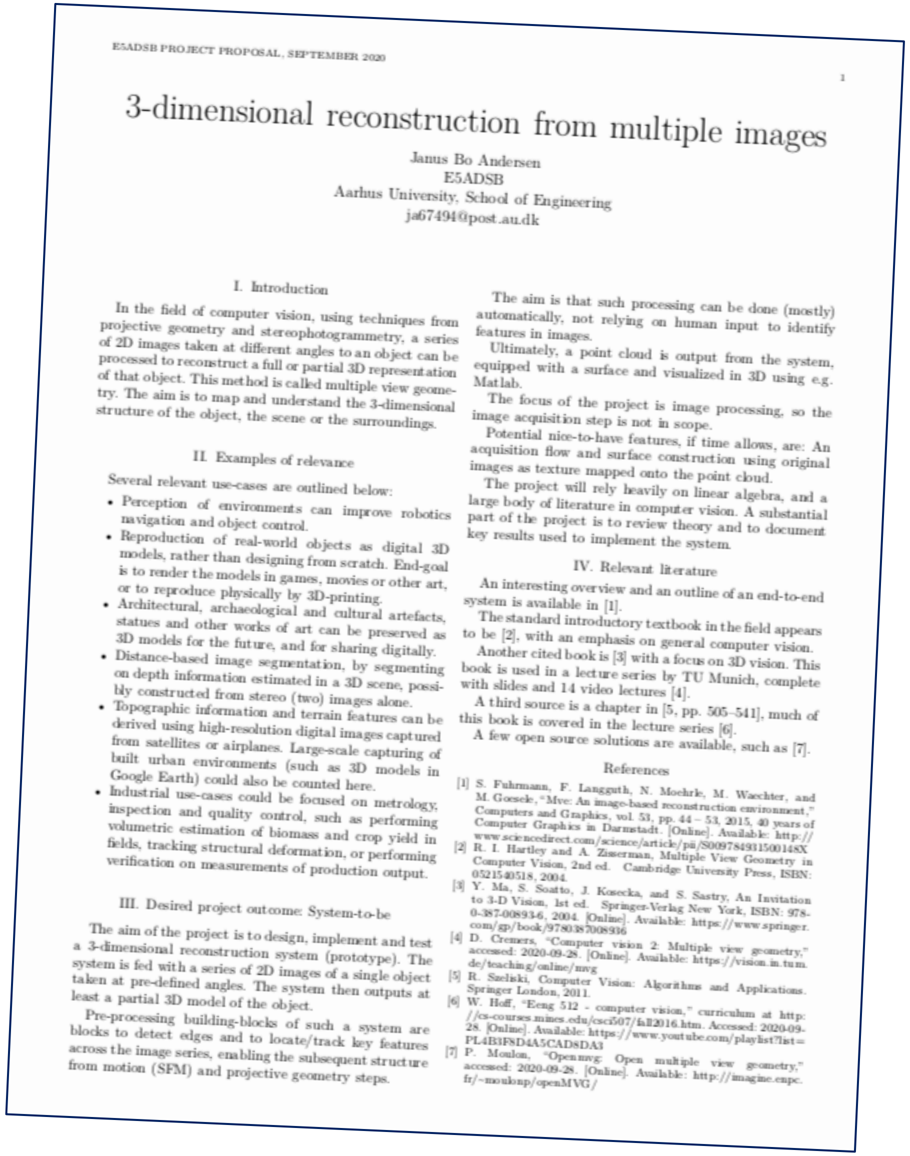
- Der tages mange billeder med stor grad af overlap
- Fælles features identificeres på tværs af (delmængder) af billedserien
    - Features trackes fra billede til billede
    - LS-agtig fitting eller RANSAC-fitting for at bygge en point-cloud model af struktur



DISPOSITION TIL PROJEKT

Forventer nogenlunde samme forløb som i denne præsentation:

- Dokumentation og eksperimenter med projektiv geometri
- Dokumentation og eksperimenter med epipolær geometri og estimering af dybde med 2 2D-billeder
  - Dokumentation og eksperimenter på featuredetektion, matching/søgning på korrespondance og registrering af billeder (canny, Hough, SIFT, osv.)
- Forsøg med multiple-view og SfM-rekonstruktion af geometri baseret på datasæt fra (forventet) Ma, Soatto, Kosecká og Sastry (2010)



3-dimensional reconstruction from multiple images

Janus Bo Andersen
E5ADSB
Aarhus University, School of Engineering
ja67494@post.au.dk

I. Introduction

In the field of computer vision, using techniques from projective geometry and stereophotogrammetry, a series of 2D images taken at different angles to an object can be processed to reconstruct a full or partial 3D representation of that object. This method is called multiple view geometry. The aim is to map and understand the 3-dimensional structure of the object, the scene or the surroundings.

II. Examples of relevance

Several relevant use-cases are outlined below:

- Perception of environments can improve robotics navigation and object control.
- Reproduction of real-world objects as digital 3D models, rather than designing from scratch. End-goal is to render the models in games, movies or other art, or to reproduce physically by 3D-printing.
- Architectural, archaeological and cultural artefacts, statues and other works of art can be preserved as 3D models for the future, and for sharing digitally.
- Distance-based image segmentation, by segmenting on depth information estimated in a 3D scene, possibly constructed from stereo (two) images alone.
- Topographic information and terrain features can be derived using high-resolution digital images captured from satellites or airplanes. Large-scale capturing of built urban environments (such as 3D models in Google Earth) could also be counted here.
- Industrial use-cases could be focused on metrology, inspection and quality control, such as performing volumetric estimation of biomass and crop yield in fields, tracking structural deformation, or performing verification on measurements of production output.

III. Desired project outcome: System-to-be

The aim of the project is to design, implement and test a 3-dimensional reconstruction system (prototype). The system is fed with a series of 2D images of a single object taken at pre-defined angles. The system then outputs at least a partial 3D model of the object.

Pre-processing building-blocks of such a system are blocks to detect edges and to locate/track key features across the image series, enabling the subsequent structure from motion (SFM) and projective geometry steps.

The aim is that such processing can be done (mostly) automatically, not relying on human input to identify features in images.

Ultimately, a point cloud is output from the system, equipped with a surface and visualized in 3D using e.g. Matlab.

The focus of the project is image processing, so the image acquisition step is not in scope.

Potential nice-to-have features, if time allows, are: An acquisition flow and surface construction using original images as texture mapped onto the point cloud.

The project will rely heavily on linear algebra, and a large body of literature in computer vision. A substantial part of the project is to review theory and to document key results used to implement the system.

IV. Relevant literature

An interesting overview and an outline of an end-to-end system is available in [1].

The standard introductory textbook in the field appears to be [2], with an emphasis on general computer vision.

Another cited book is [3] with a focus on 3D vision. This book is used in a lecture series by TU Munich, complete with slides and 14 video lectures [4].

A third source is a chapter in [5, pp. 505–541], much of this book is covered in the lecture series [6].

A few open source solutions are available, such as [7].

References

[1] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, "Mve: An image-based reconstruction environment," Computers and Graphics, vol. 53, pp. 44 – 53, 2015, 40 years of Computer Graphics in Darmstadt. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S009784931500148X
[2] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
[3] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, An Invitation to 3-D Vision, 1st ed. Springer-Verlag New York, ISBN: 978-0-387-00893-6, 2004. [Online]. Available: https://www.springer.com/gp/book/9780387008936
[4] D. Cremers, "Computer vision 2: Multiple view geometry," accessed: 2020-09-28. [Online]. Available: https://vision.in.tum.de/teaching/online/mvg
[5] R. Szeliski, Computer Vision: Algorithms and Applications. Springer London, 2011.
[6] W. Hoff, "Eeng 512 - computer vision," curriculum at http://cs-courses.mines.edu/csci507/fall2016.htm. Accessed: 2020-09-28. [Online]. Available: https://www.youtube.com/playlist?list=PL4B3F8D4A5CAD8DA3
[7] P. Moulon, "Openmvg: Open multiple view geometry," accessed: 2020-09-28. [Online]. Available: http://imagine.enpc.fr/~moulonp/openMVG/

AARHUS
UNIVERSITET