Semantic annotation of urban scenes: Skyline and window detection

Speaker: Tjerk Kostelijk

Supervisors:
Isaac Esteban
Prof. dr ir Frans C. A. Groen

July 4, 2012

Outline

Experiment

Experiment

420px

Did you see?

- building

Did you see?

- building
- tree

Did you see?

- building
- tree
- bicycle

Did you see?

- building
- tree
- bicycle
- street lamb

Did you see?

- building
- tree
- bicycle
- street lamb
- red car

Did you see?

- building
- tree
- bicycle
- street lamb
- red car
- blue car

Did you see?

- building
- tree
- bicycle
- street lamb
- red car
- blue car
- brand of the car?

Experiment

Experiment

420px

Q

- Why are we so good at object/depth recognition?

Q

- Why are we so good at object/depth recognition?
- How can we apply vision to a computer system?

Q

- Why are we so good at object/depth recognition?
- How can we apply vision to a computer system?
  - *Computer Vision*

Outline

Human perception

- depth cue, binocular disparity

Human perception

- depth cue, binocular disparity
- Classify objects: feature detection

Human perception

- depth cue, binocular disparity
- Classify objects: feature detection
  - straight lines

Human perception

- depth cue, binocular disparity
- Classify objects: feature detection
  - straight lines
  - right angles

Where is my research about?

- Annotation of urban scenes

Where is my research about?

- Annotation of urban scenes
  - Skyline detection

Where is my research about?

- Annotation of urban scenes
  - Skyline detection

Where is my research about?

- Annotation of urban scenes
  - Skyline detection

Outline

Application examples of annotation of urban scenes

- 3D city models

Application examples of annotation of urban scenes

- 3D city models
- Driving simulation

Application examples of annotation of urban scenes

- 3D city models
- Driving simulation
- Augmented reality

Application examples of annotation of urban scenes

- 3D city models
- Driving simulation
- Augmented reality
- Building recognition

Application examples of annotation of urban scenes

- 3D city models
- Driving simulation
- Augmented reality
- Building recognition
- Analysis building deformation

Application examples of annotation of urban scenes

- 3D city models
- Driving simulation
- Augmented reality
- Building recognition
- Analysis building deformation
  - 'noord-zuidlijn'

Skyline detection application example

- Horizon detection for Unmanned Air Vehicles

width=420px

Skyline in urban scenes

- Canny edge detection

Skyline in urban scenes

- Canny edge detection
- Result: Binary image (edge or no edge)

Skyline in urban scenes

- Canny edge detection
- Result: Binary image (edge or no edge)
- top sharp edge assumption
  "The first sharp edge (seen from top to bottom) in the image represents the skyline."

Skyline detection algorithm

- The image is sliced in #w pixelcolumns

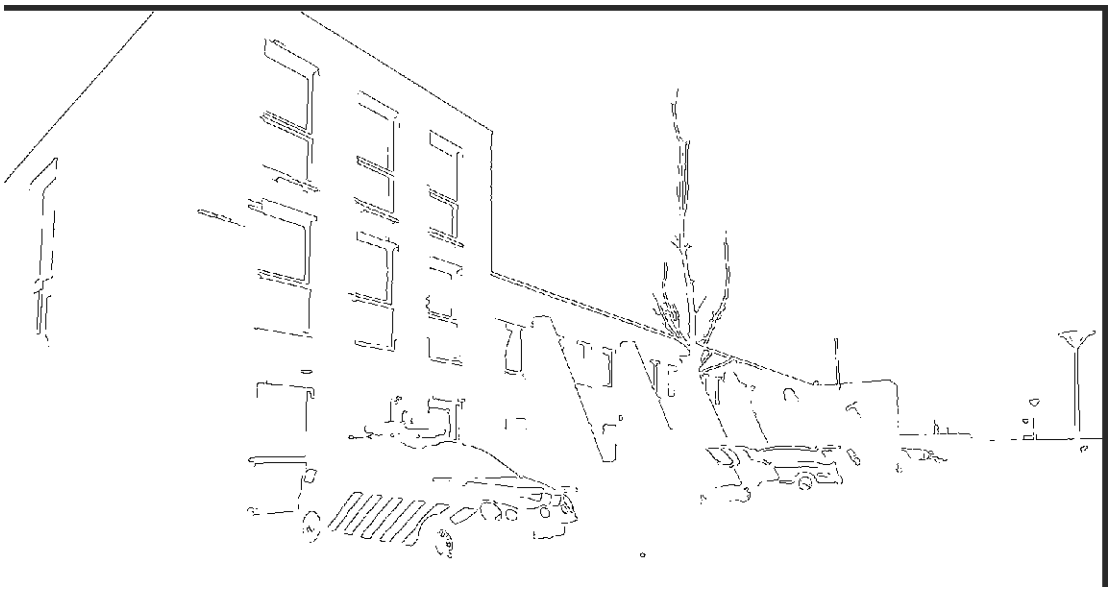
Skyline detection algorithm

- The image is sliced in #w pixelcolumns
- Each column present #h binary edge values (edge or no edge)

Skyline detection algorithm

- The image is sliced in #w pixelcolumns
- Each column present #h binary edge values (edge or no edge)
- y-location of the first edge value is stored

Skyline detection Result



Hypothesis based skyline detection, assumption

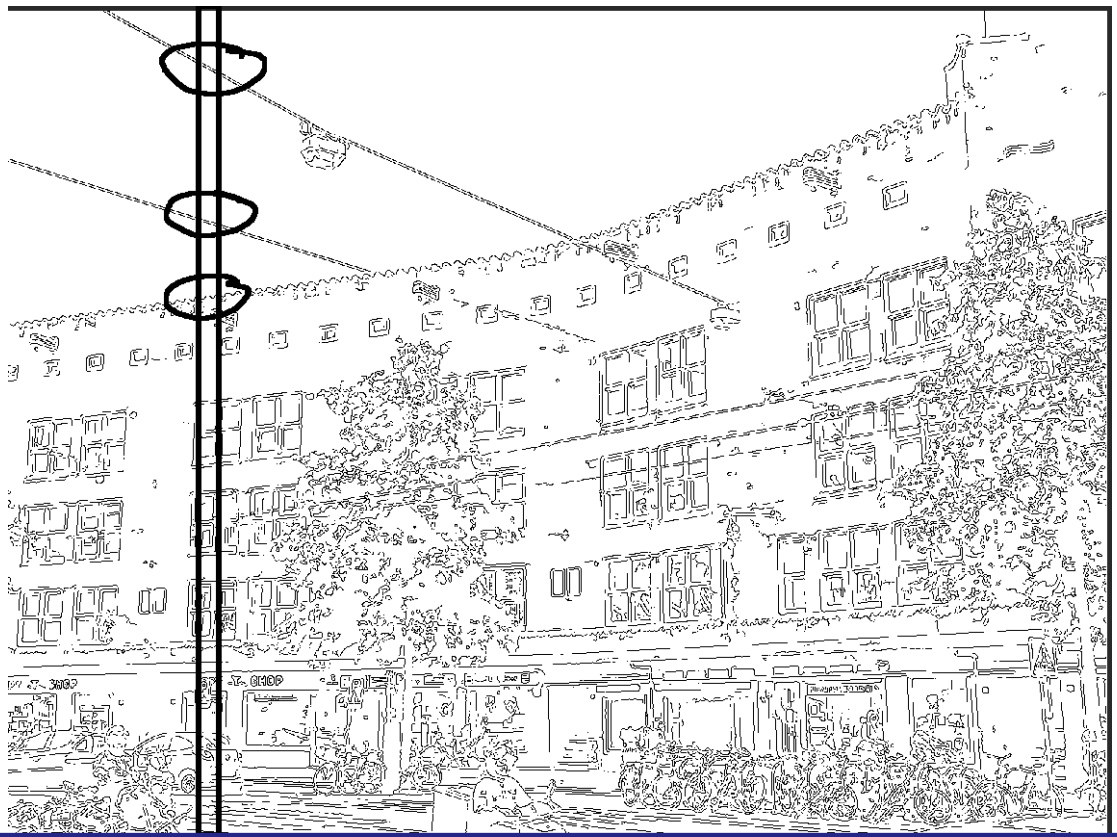
"The first sharp edge (seen from top to bottom) in the image represents the skyline."

- Example of a scene where this assumption is violated

Hypothesis based skyline detection, assumption

- Change of assumption

"The skyline is part of the first n sharp edges (seen from top) (e.g. $n = 3$)



Hypothesis based skyline detection, algorithm

- Use exsisting column based approach

Hypothesis based skyline detection, algorithm

- Use exsisting column based approach
- Generate *n* hypothesis

Hypothesis based skyline detection, algorithm

- Use exsisting column based approach
- Generate *n* hypothesis
- classify hypothesis with additional info

Hypothesis based skyline detection, algorithm

- Use exsisting column based approach
- Generate *n* hypothesis
- classify hypothesis with additional info
  - texture

Hypothesis based skyline detection, algorithm

- Use exsisting column based approach
- Generate *n* hypothesis
- classify hypothesis with additional info
  - texture
  - color

Hypothesis based skyline detection, algorithm

- Use exsisting column based approach
- Generate *n* hypothesis
- classify hypothesis with additional info
  - texture
  - color
  - height variation

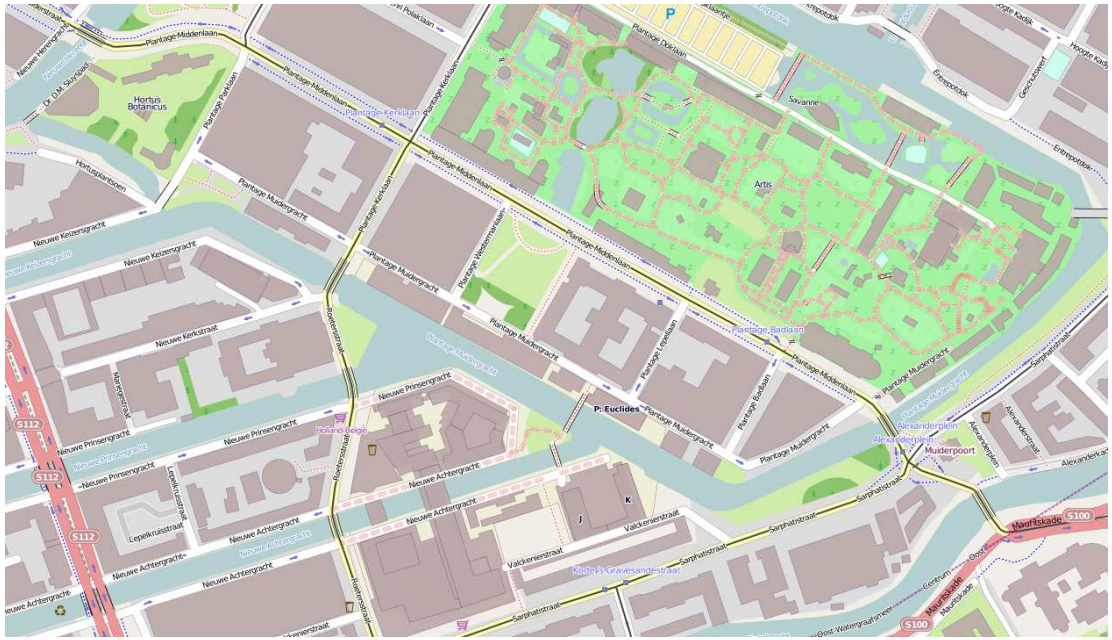
Expected result on hypothesis classification based on color

Expected result on hypothesis classification based on heigth variation

Outline

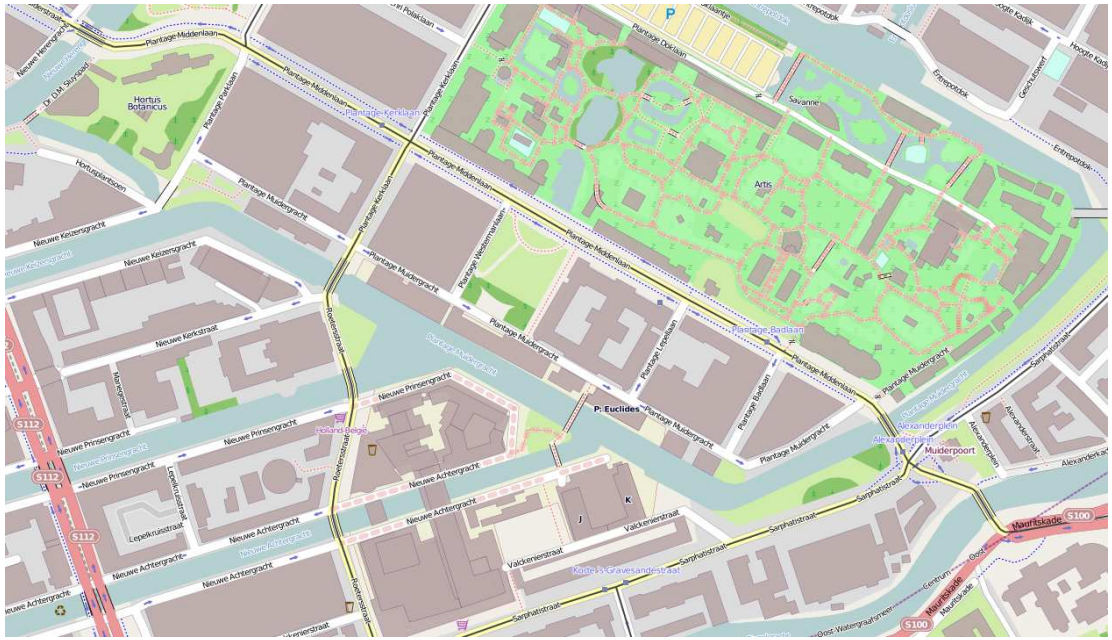
Overview

- Create (top view) 2D model of the scene using *Openstreetmap*



Overview

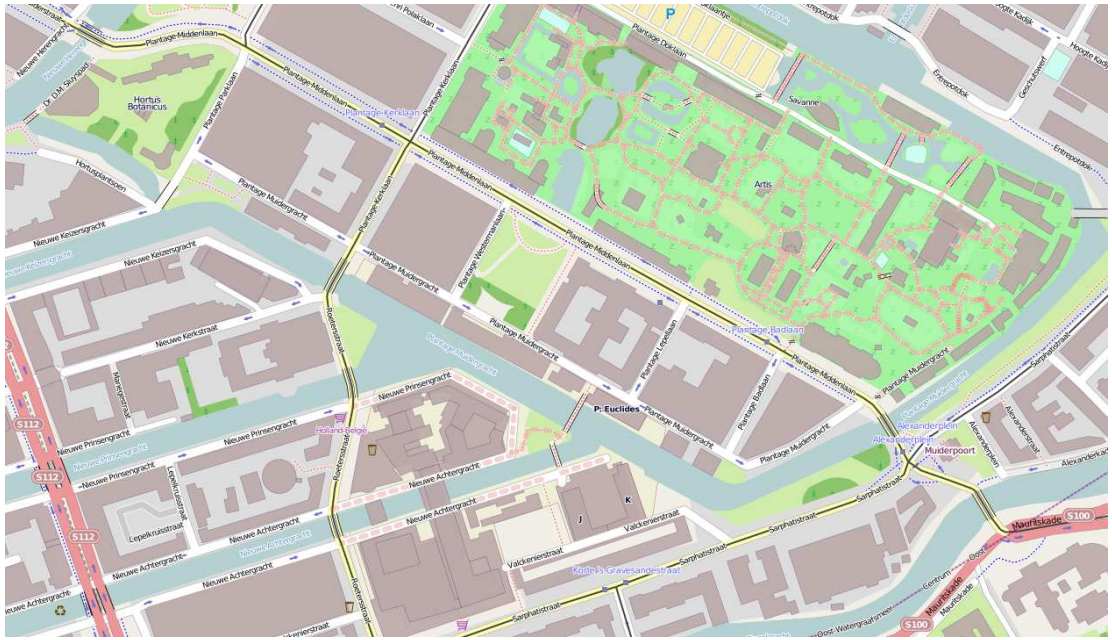
- Create (top view) 2D model of the scene using *Openstreetmap*



- Align 2D model with the scene using *FIT3D toolbox*

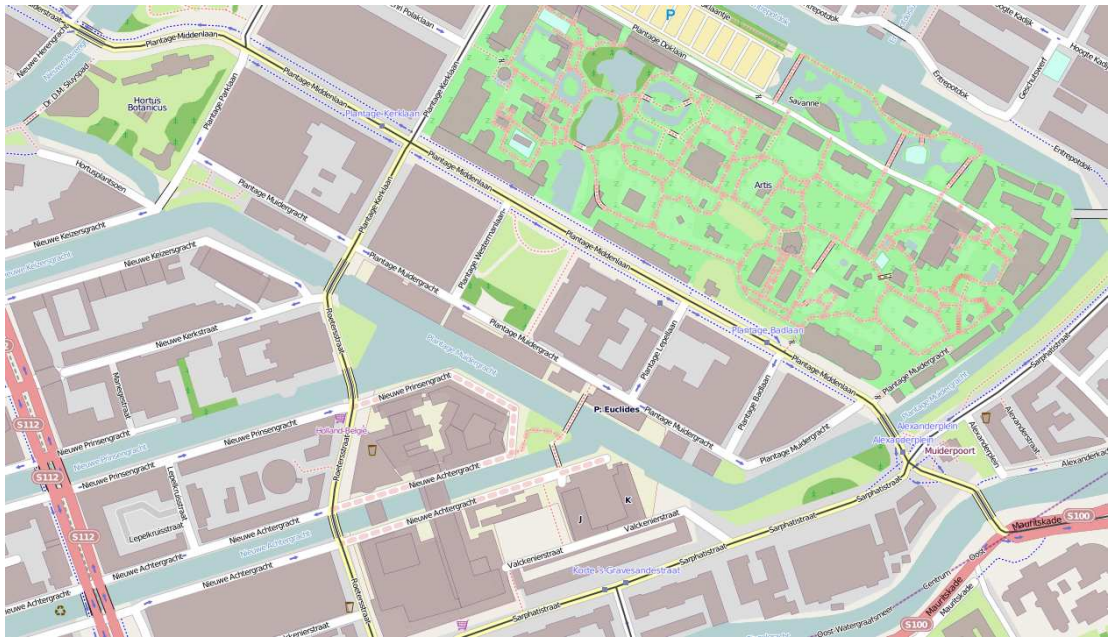
Overview

- Create (top view) 2D model of the scene using *Openstreetmap*



- Align 2D model with the scene using *FIT3D toolbox*
- Transform the 2D model to a 3D model by extending the walls

Extract 2D model



Align 2D model

- *FIT3D toolbox*

(a)

Align 2D model

- *FIT3D toolbox*
  - Input: sequence of images that contain different views of the building

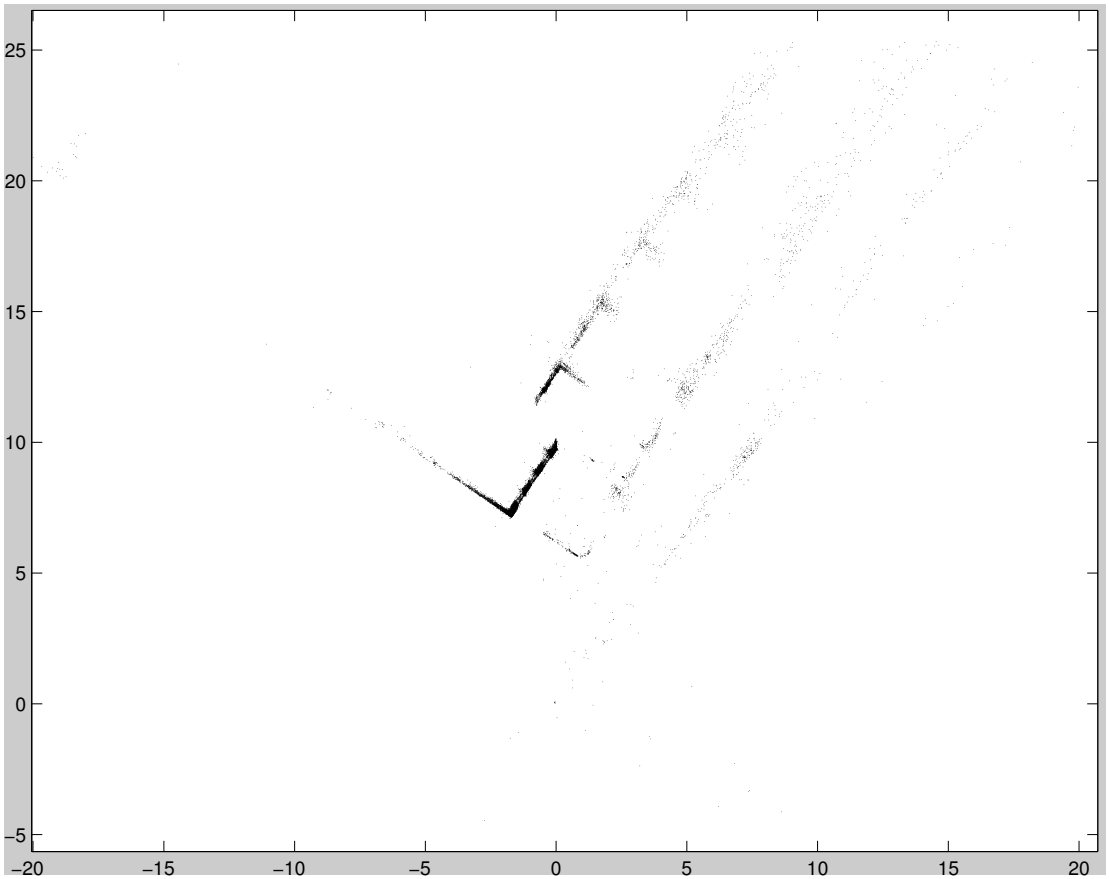
(c)

Align 2D model

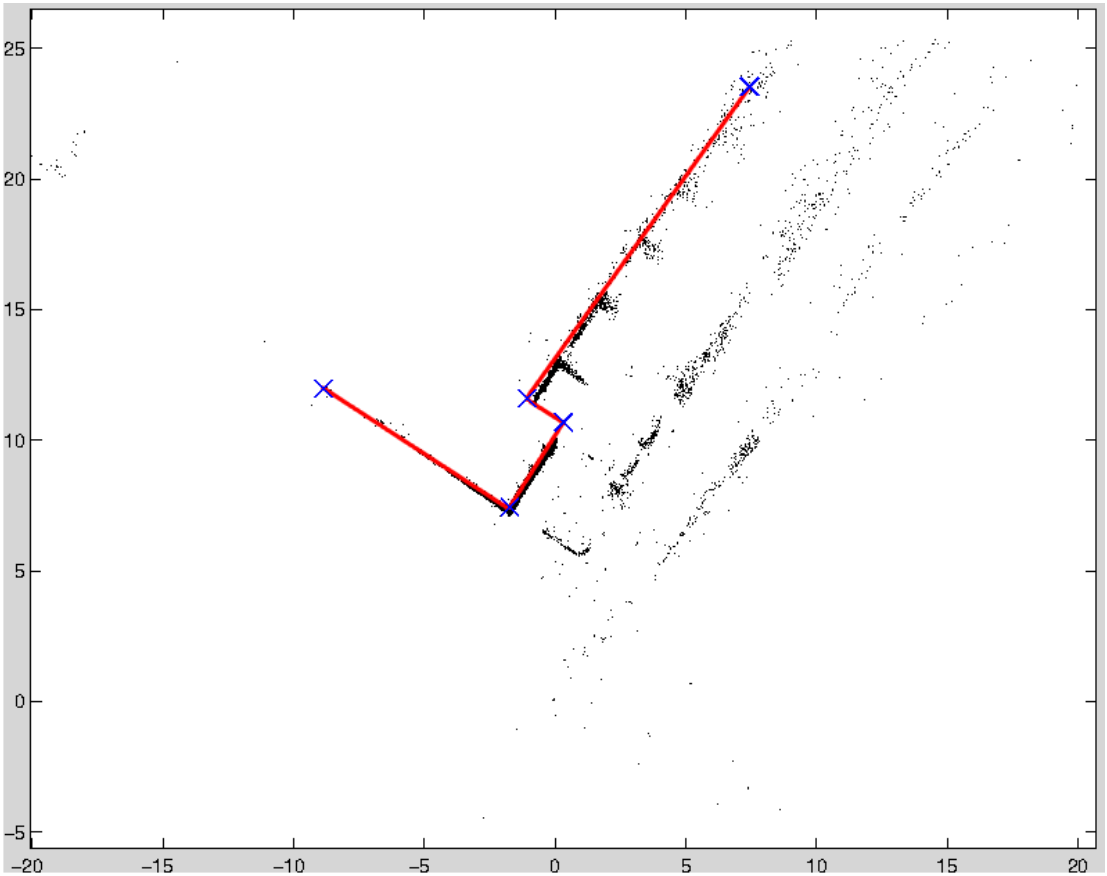
- *FIT3D toolbox*
  - Input: sequence of images that contain different views of the building
  - Output: a 3D point cloud of the building

(e)

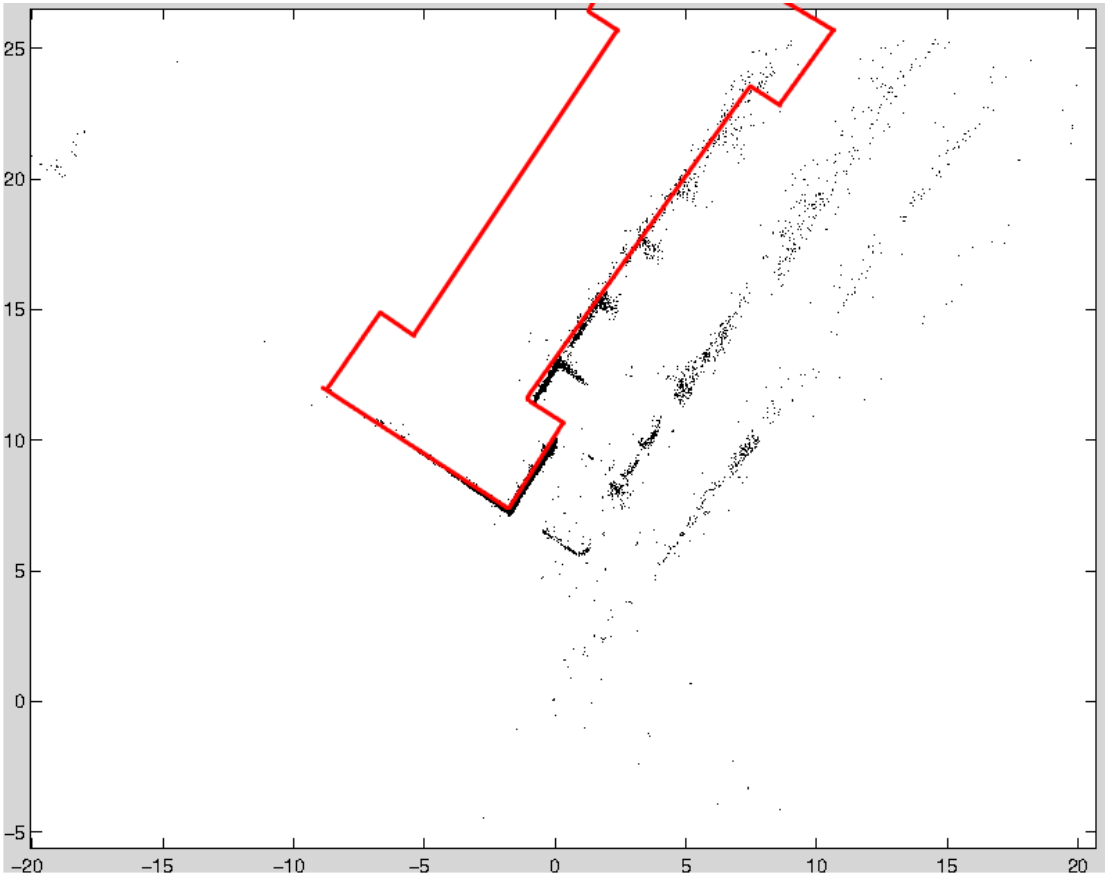
Align 2D model



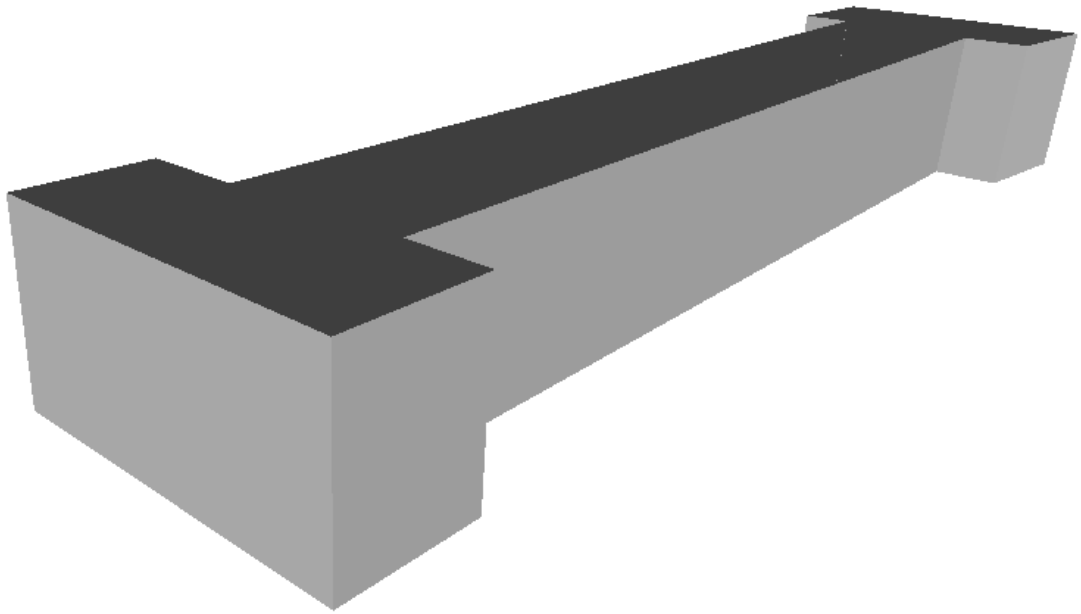
Align 2D model



Align 2D model

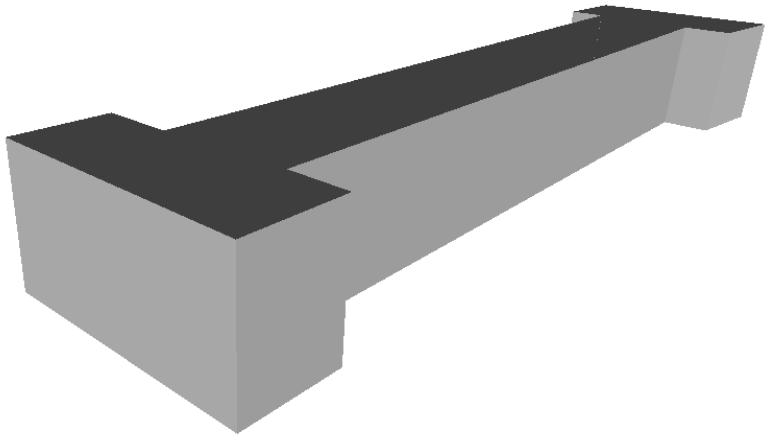


Results



Wall height estimation

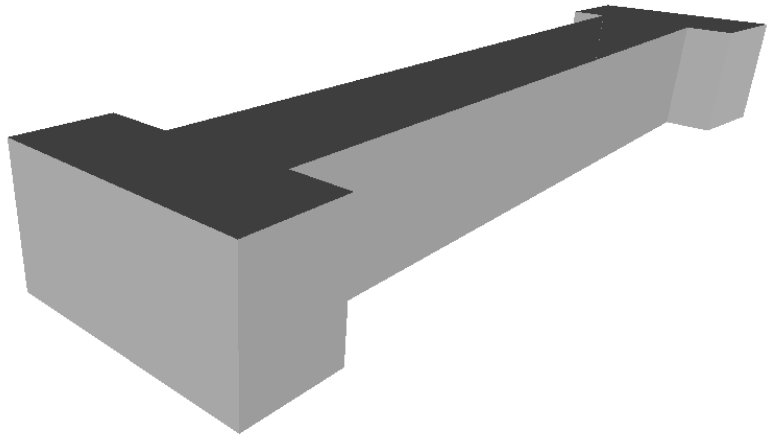
- Wall heights of 3D model are not accurate



(g)

Wall height estimation

- Wall heights of 3D model are not accurate



(i)

Assumptions

Straight lines in the skyline are likely to come from the building contour Flat roof assumption -¿ building contour is equal to upper side of building walls

Extracting line segments

- Output of skyline detector

Extracting line segments

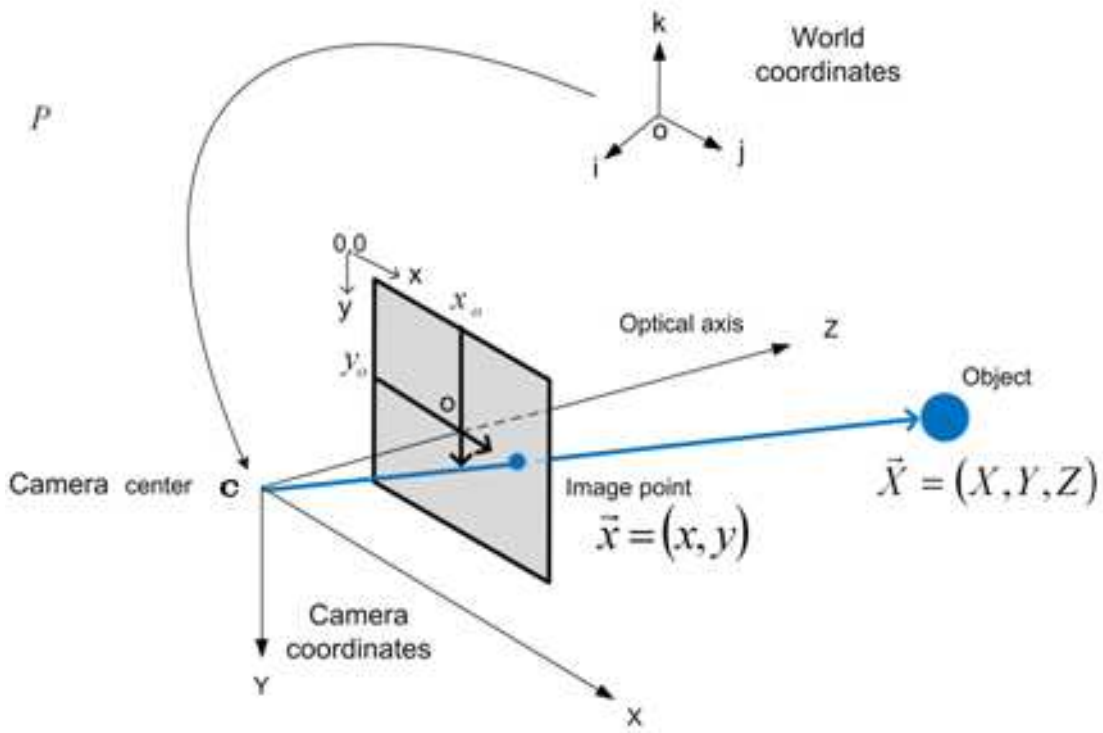
- Output of skyline detector
- Hough transform

Project to the 3D model

- Estimate wall heights

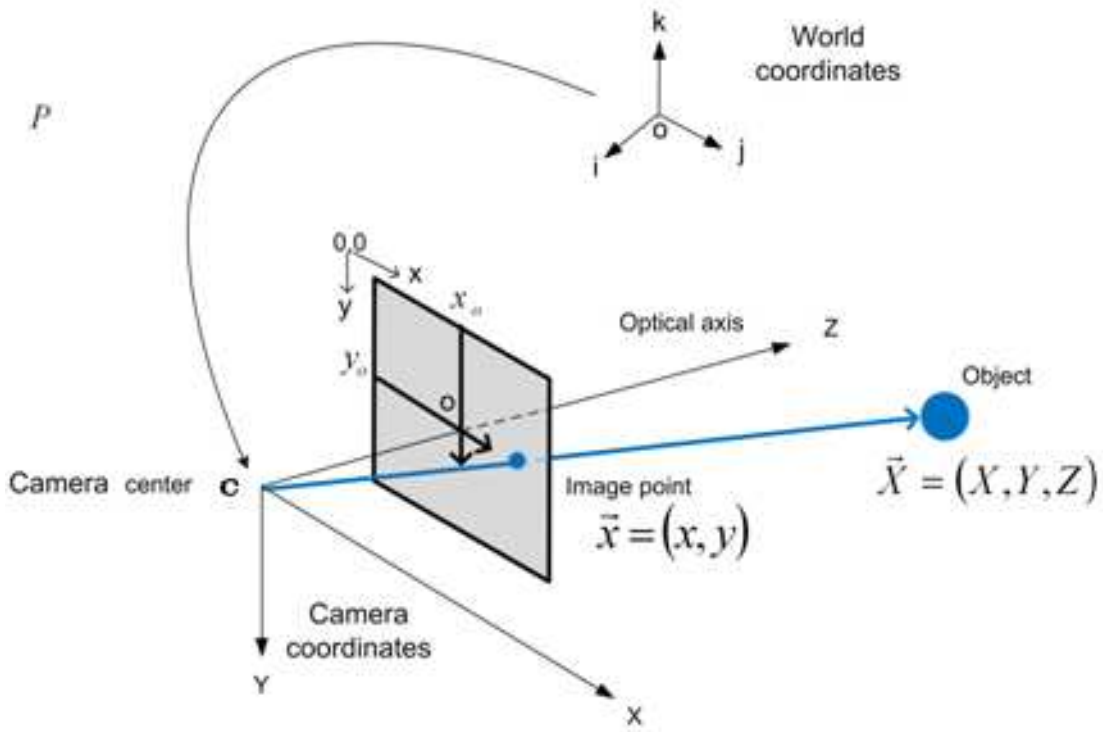
Project to the 3D model

- Estimate wall heights
- project line segments to specific walls of 3D model

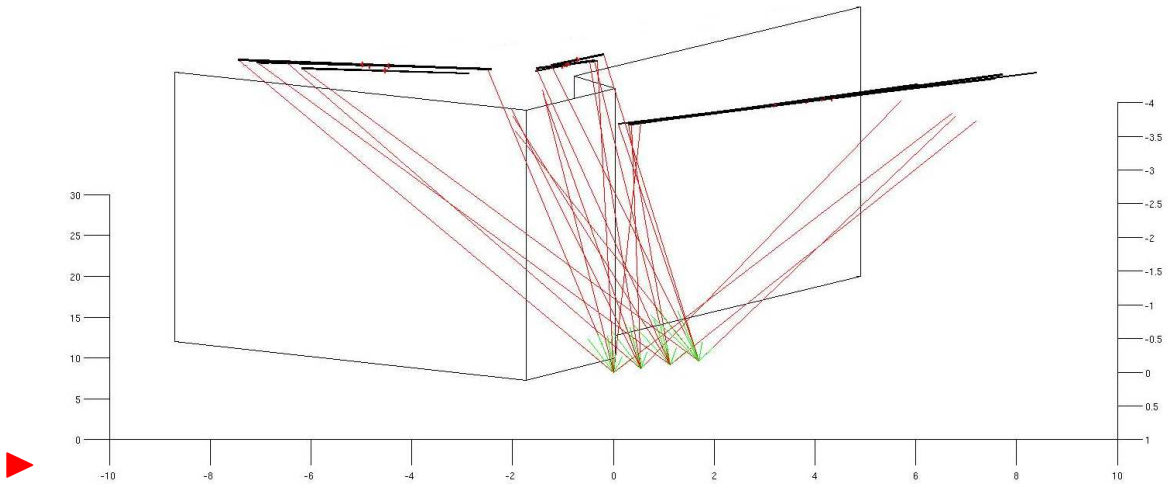


Project to the 3D model

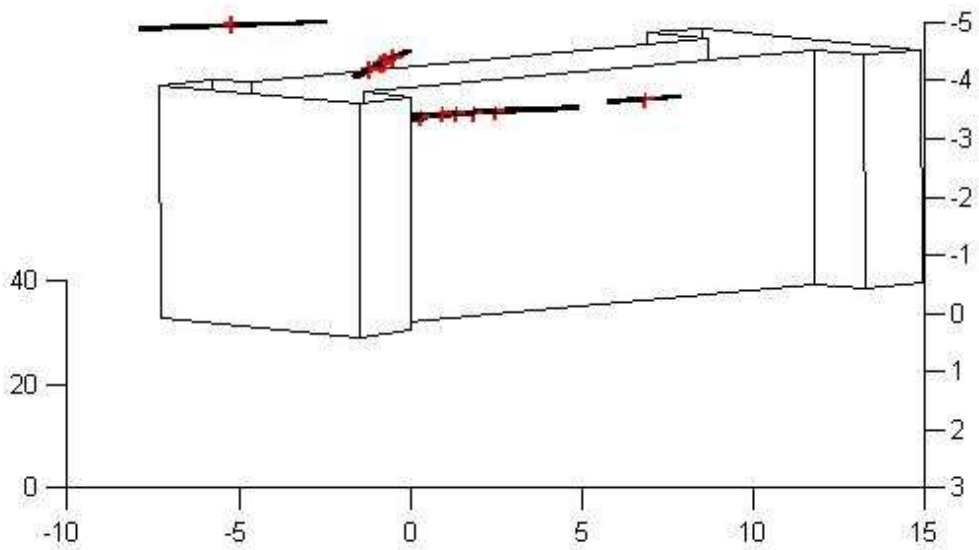
- Estimate wall heights
- project line segments to specific walls of 3D model



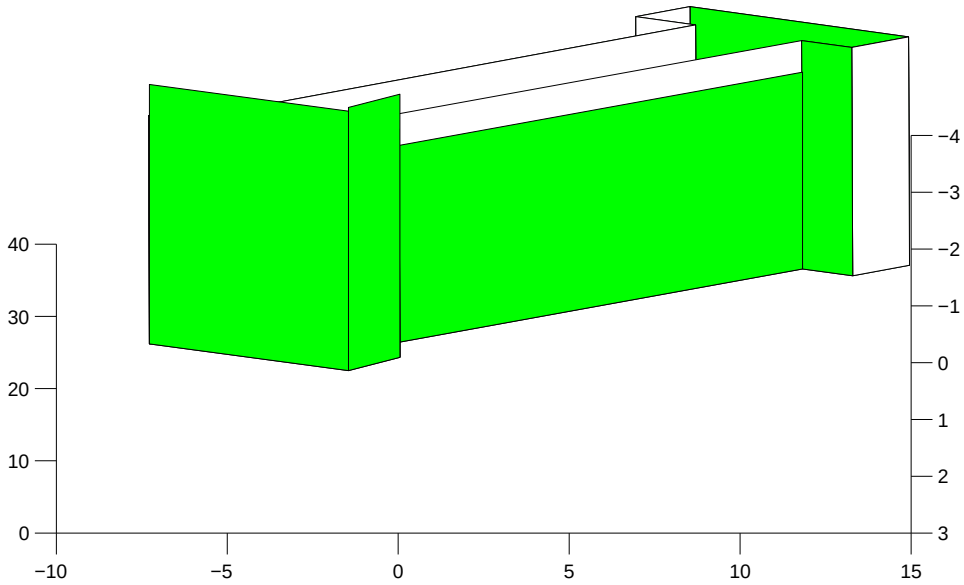
Results 1/3



Results 2/3



Results 3/3



Improved 3D model0.6

Future research

Overview

Outline Window detection

- Method I: Connected corner approach

Outline Window detection

- Method I: Connected corner approach
  - invariant to viewing direction

Outline Window detection

- Method I: Connected corner approach
  - invariant to viewing direction
- Facade rectification

Outline Window detection

- Method I: Connected corner approach
  - invariant to viewing direction
- Facade rectification
- Method II: Histogram based approach

Outline Window detection

- Method I: Connected corner approach
  - invariant to viewing direction
- Facade rectification
- Method II: Histogram based approach
  - rectified facades

Outline Window detection

- Method I: Connected corner approach
  - invariant to viewing direction
- Facade rectification
- Method II: Histogram based approach
  - rectified facades
  - uses Histograms of Houghlines

Method I: Connected corner approach

Situation

- Window frames

Situation

- Window frames
- Difference color window frame, color glass

Situation

- Window frames
- Difference color window frame, color glass
- Produce edges in horizontal and vertical direction

The idea

- Connected corner

The idea

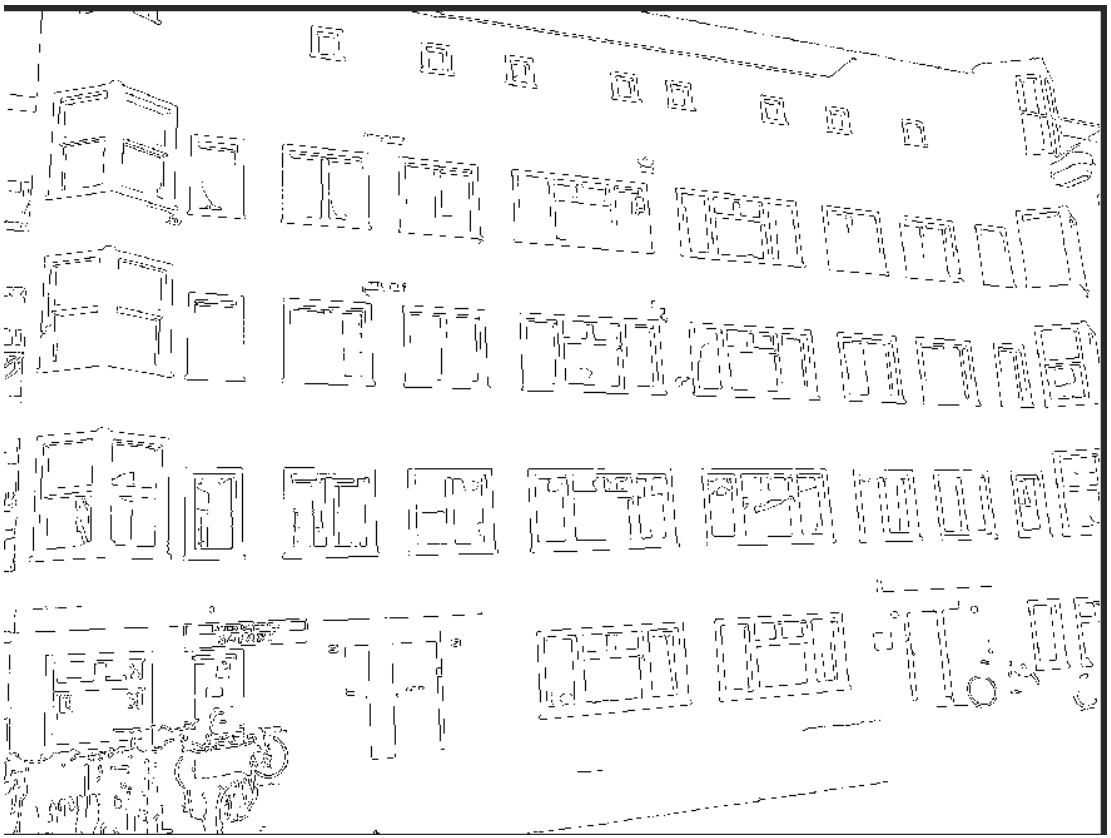
- Connected corner
- *"horizontal and vertical edges that come from the same (sub)window frame share a corner"*

Edge detection and Houghline extraction

- Canny edge detector
- Two groups of straight lines (Houghlines)
  - Horizontal, $\theta = [-30..0..30)$ degrees
  - Vertical, $\theta = [80..90..100)$ degrees
- Why the use of angle ranges?
  - Camera not exactly upright
  - Facade view contains an angle, perspective distortion

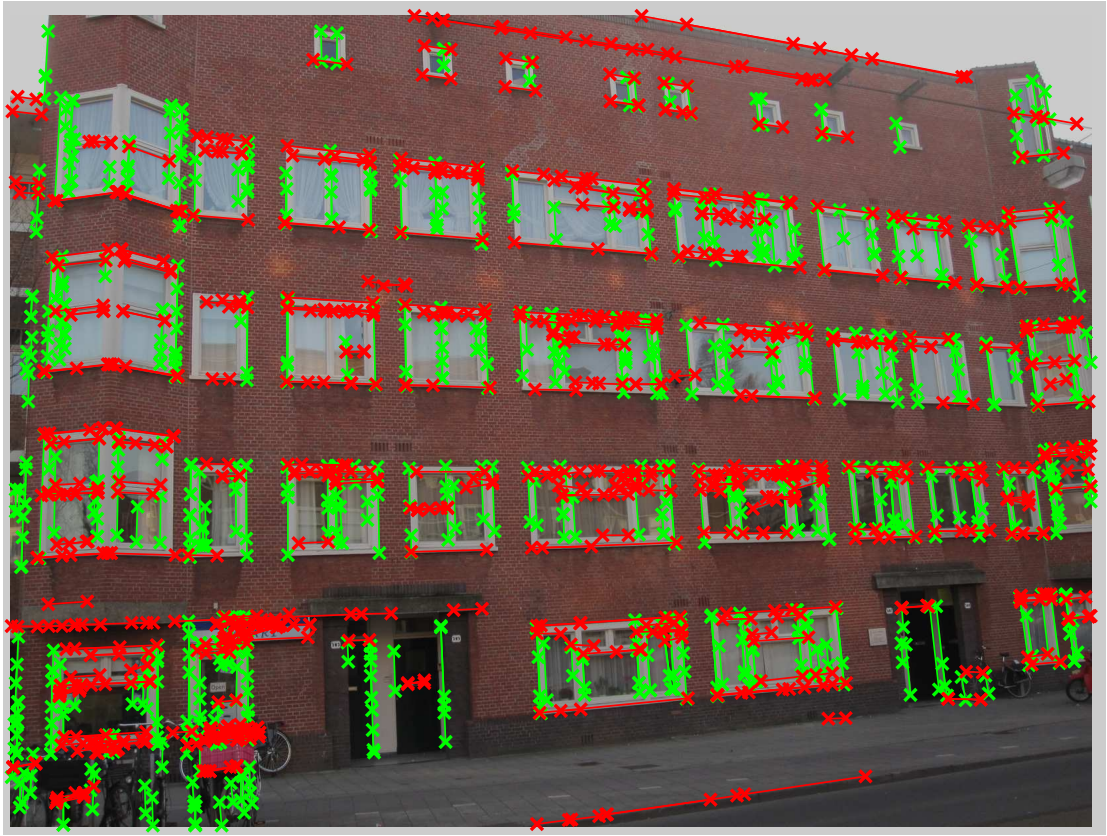
Original

Edge detection



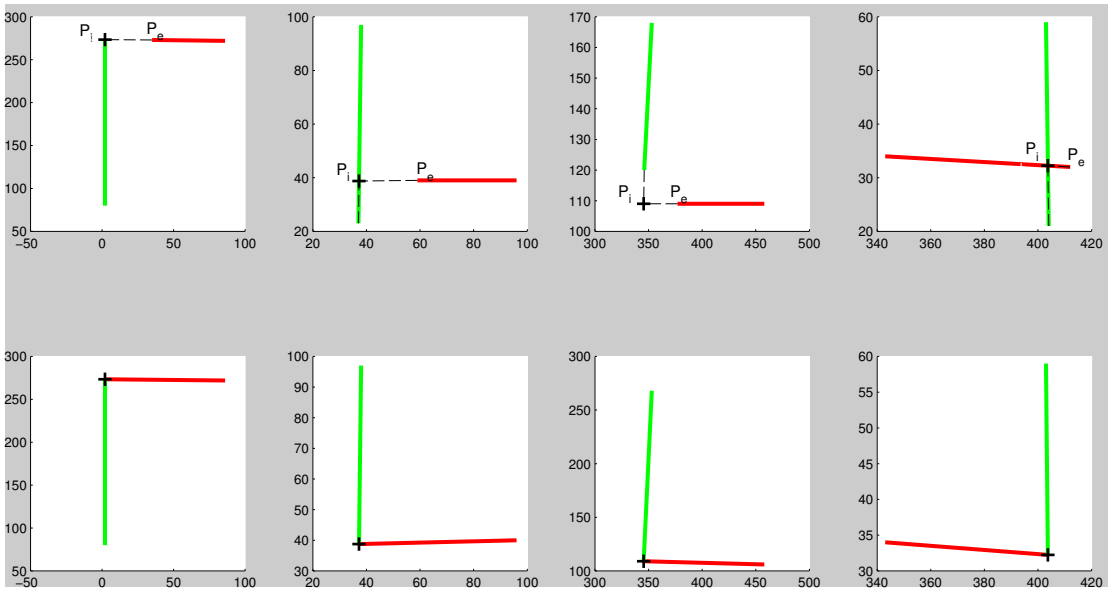
Edge detection0.45

Result of θ constrained Hough transform



Connected corner

- Connected corner:a horizontal and vertical edge that share the same corner e.g. L
- Algorithm: If line segments endpoints are close enough a connected corner is formed
- Tolerate gab or extension



Connected Corners

Window area extraction

- Mirror from diagonal through endpoints
- L shape -¿ Quadrangle window area

Results

Conclusion

- The skyline deteciton algorithm
  - is simple and has a low complexity
  - works wel under the assumption skyline is upper edge
  - can provide a set of hypothesis which should be evaluated using andditional features

Future research

- L-shapes, U-shapes
- analysis of substructure of windows
  - define a contained in relations
  - cluster connected corner on location and length
    - maximum intercluster distance correlates with size subwindow
    - assume nr of subwindows to estimate size

Method II: window detection

- Assumes that the windows are
  - orthogonal
  - aligned

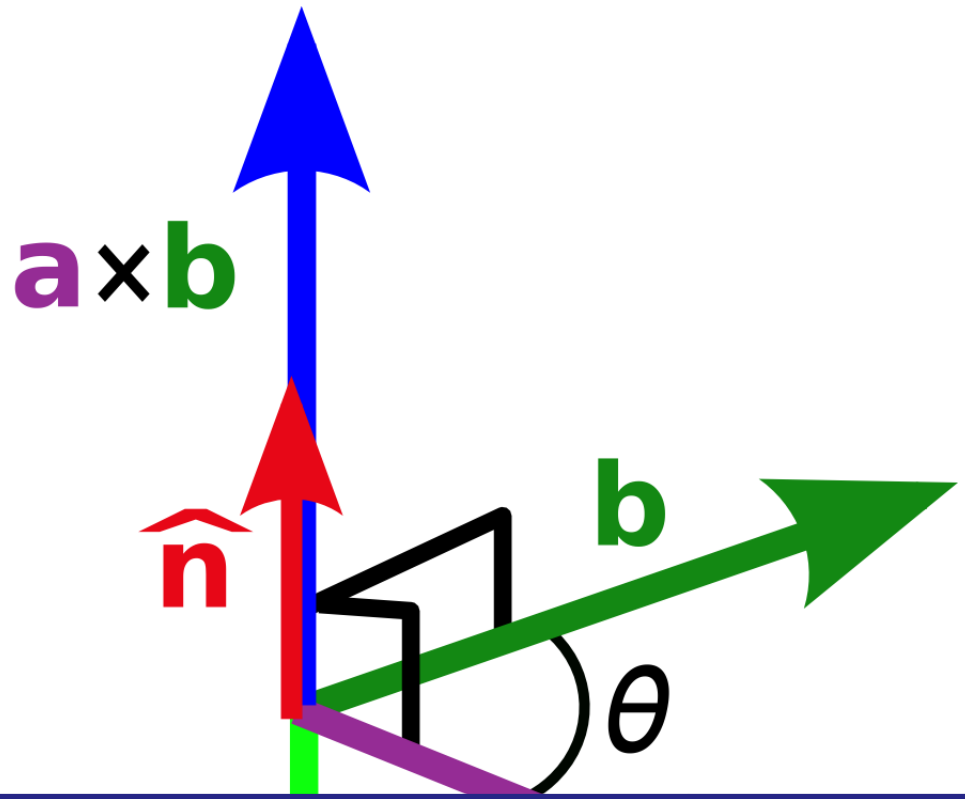
Aligned but not orthogonal

Rectified facade

Dataset: Anne1, Rectified image0.45

Facade rectification

- 3D model gives plane that corresponds to wall
- wall produces a normal vector b
- camera's heading, $\vec{a}$
- rotation matrix R is calculated and applied



Unrectified facade

Rectified facade

Dataset: Anne1, Rectified image0.45


SAFARI


SAFARI

Extracting the window alignment

- Window alignment line
  - *"A horizontal or vertical line that aligns multiple windows"*
- Alignment lines provide grid of blocks
- Window non-window areas

Window alignment lines example

Outline

osgviewer demo of 3d point cloud and images demo of matlab

progje

Questions

Questions?