3D City Model Aided Change Detection from Aerial Stereo-Images

Kaixuan Zhou
PhD student from GRS, CiTG

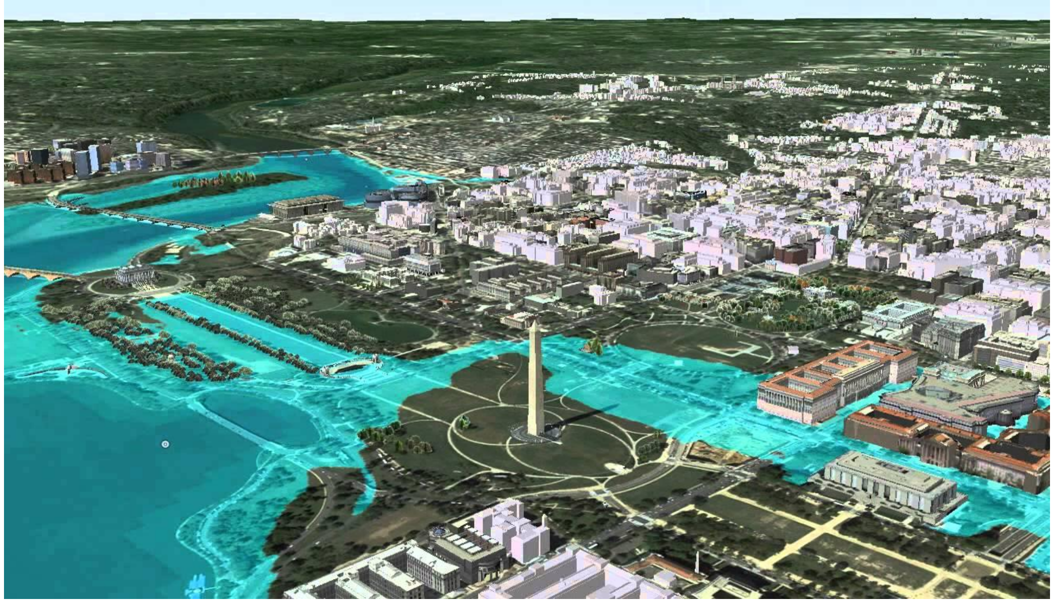
Content

- Introduction
- Objective
- General Schema
- Shadow

Introduction

•Up-to-date 3D city model for disaster management

- Disaster simulation
- 3D visualization- Visual reality
- Reference for rescue



Climate Decision flood simulation of Washington, DC.
Source: http://climatedecision.net/

Introduction

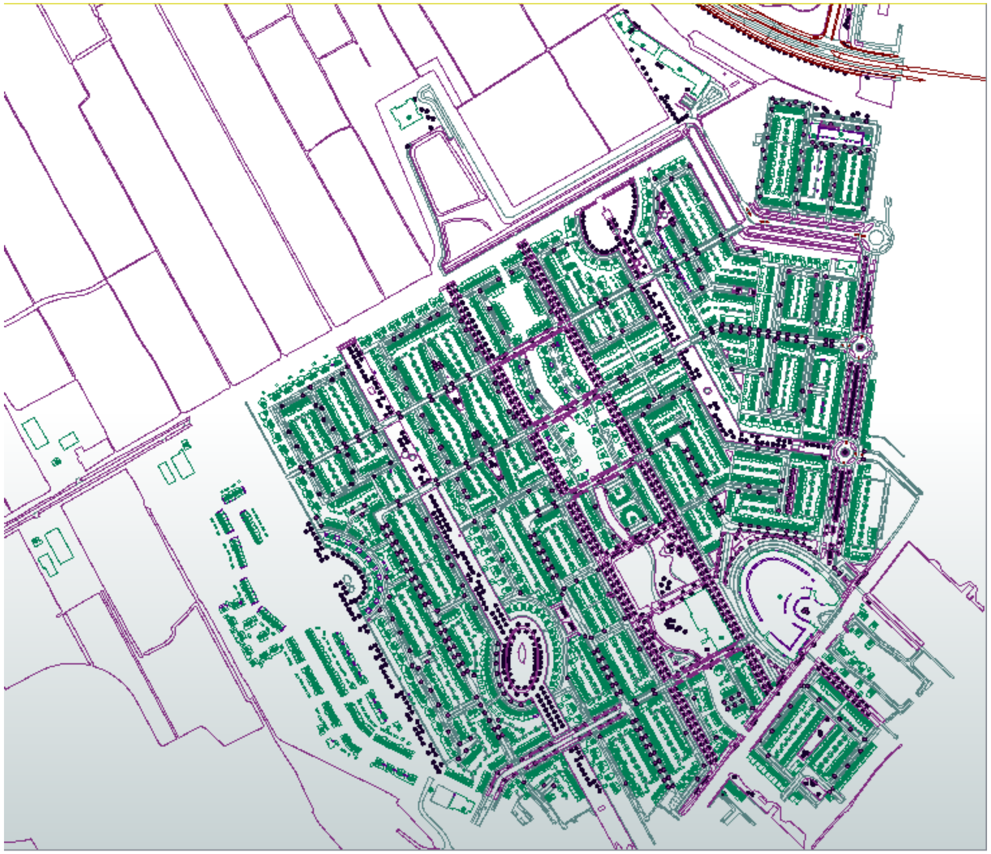
- **Maps4society**
  - Generation of Up-to-date 3D city models for water management
  - 3D city model is for run-off modelling
- **Water management**
  - Buildings
  - Vegetation
  - Streets

Introduction

- **3D city model in T_0--Base map**
  - BGT –update 1-2 years
  - AHN2 –update 5-10 years

- **3D city model in T_1-- Update annually**
  - Aerial photo
  - Stereo from Image matching

- **Change detection**
  - BGT and AHN2- ITC, UT
  - 3D Model and VHR Image- GRS, TUD



Introduction

- **3D city model in T_0--Base map**
  - BGT –update 1-2 years
  - AHN2 –update 5-10 years

- **3D city model in T_1-- Update annually**
  - Aerial photo
  - Stereo from Image matching

- **Change detection**
  - BGT and AHN2- ITC, UT
  - 3D Model and VHR Image- GRS, TUD

Objectives

- Naïve approach- 3D model objects vs 3D image object
  - 3D objects are semantically and geometrically inconsistent

  - Image classification- rule, unsupervised, supervised way
    - Case dependent thresholds
    - Strong assumptions
    - Manual work on training samples

- 3D model
  - Rich information- Prior knowledge of interpreting stereo-image or change
    - Semantic and segments– Give segmentation and label to image
    - Contexture – adjacency of building
    - Geometry and height – building’s elongation, concavity
    - Shadow

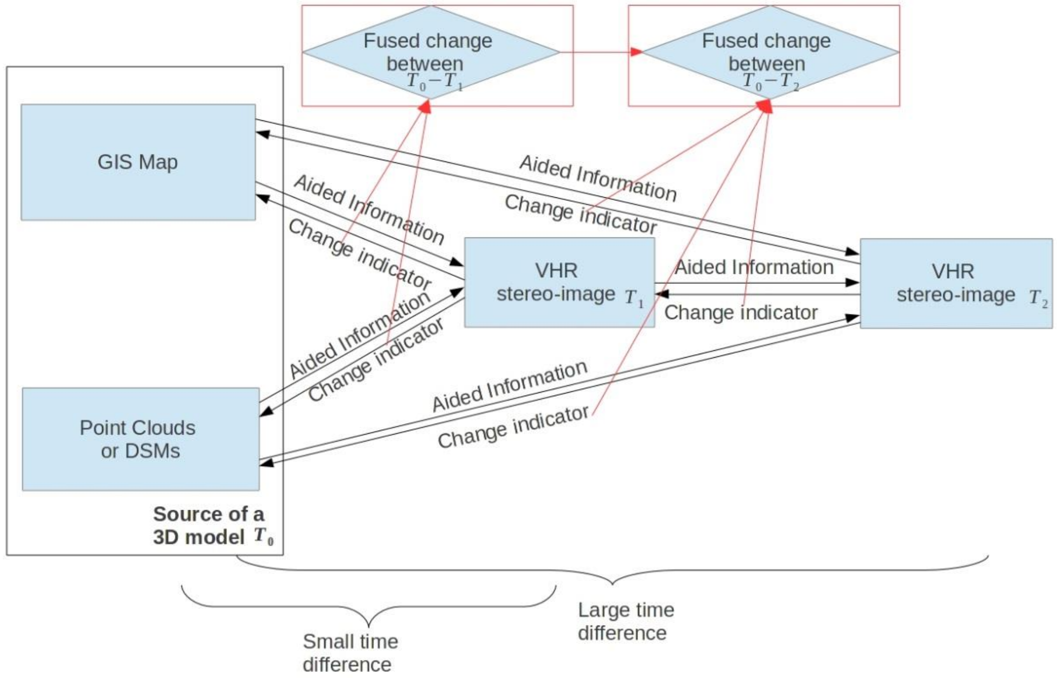
- Old image– spectral change indicator

Objectives

• Improve the accuracy of change detection by fully using information from 3D city model and the VHR image

•Create a change detection schema to be applicable in any other urban areas by using information in existing 3D model

General Schema



Shadow

•First try on using 3D information

•Serious Problem of VHR

•False alarm
when two images are compared

•Change indicator
When compared with reconstructed shadow
From 3D model

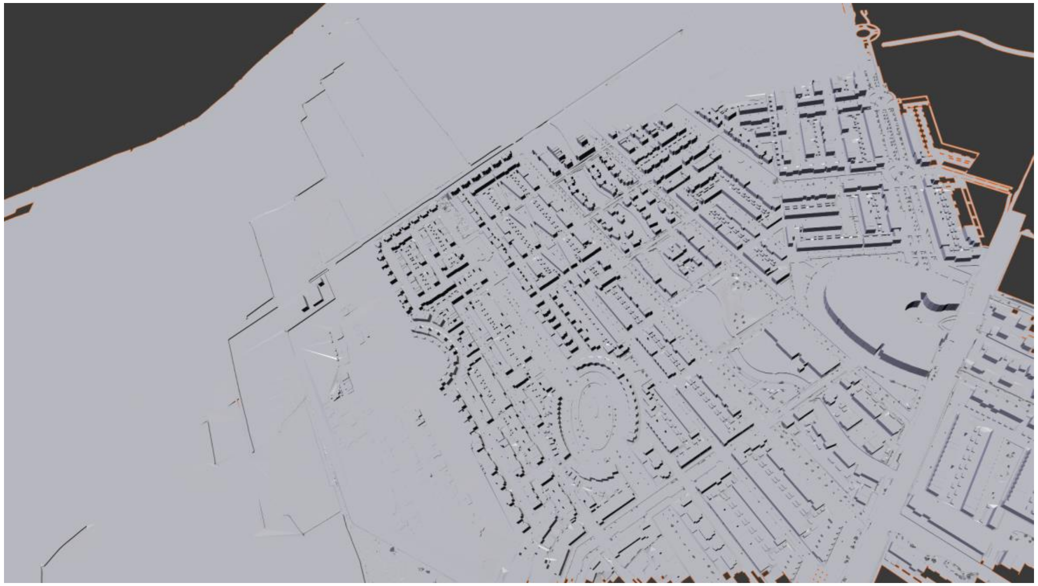
Shadow

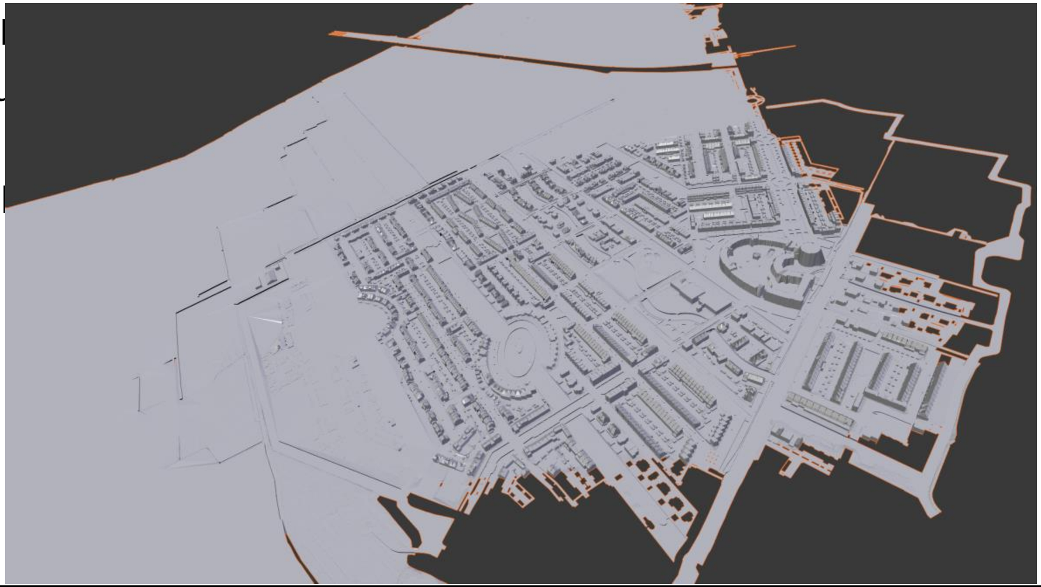
- Accurate 3D city model can reconstruct perfect shadow if no change in reality
  - 3D city model
    - LoD1 model from 3dfier- TUDelft
    - LoD2 model from 3DIMGeo -UT
  - Shadow reconstruction
    - Shadow mapping
    - Ray tracing

Shadow

- 3D city model-Mesh
  - BGT + AHN2
  - Simplified
  - Less storage
  - Accurate(in theory)

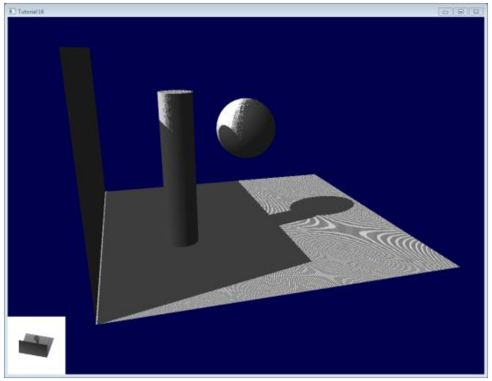
- Comparison between two software
  - 3dfier – seamless ground +Lo
  - 3DIMGeo – not seamless grou
  - Trees are not good
  - Ground from 3dfier+ LoD2 bui

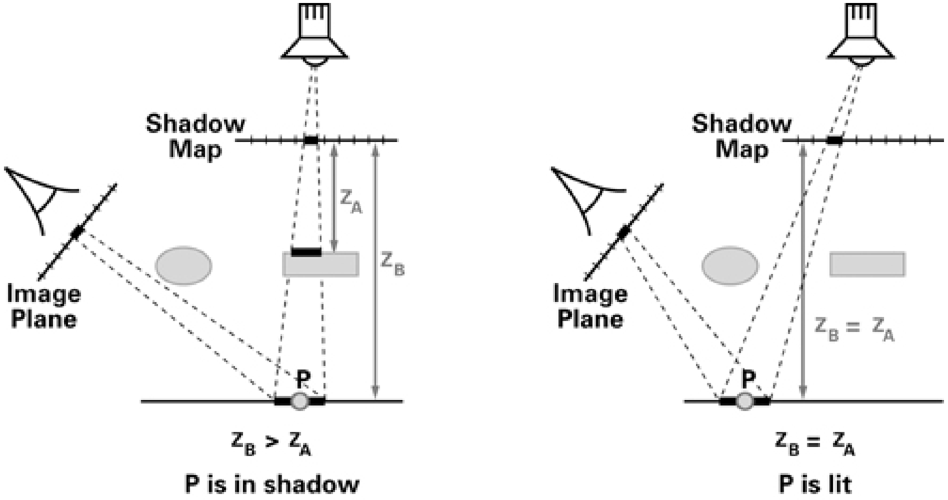




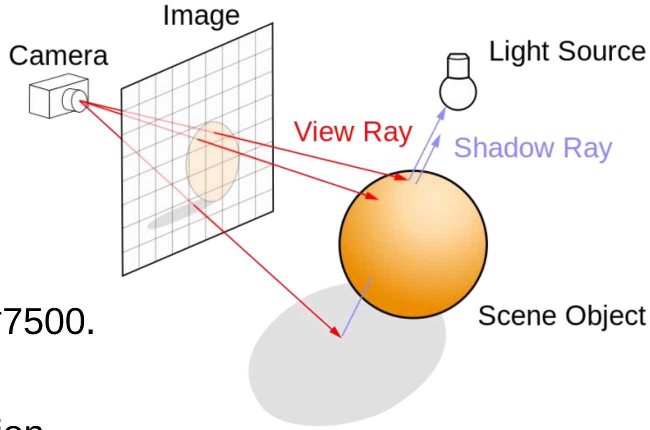
Shadow

- Shadow mapping
  - Fast but Not always correct
  - Shadow acne
  - Jagged shadow





- Raytracing
  - Straightforward and Accurate
  - Recursive trace
  - Slow



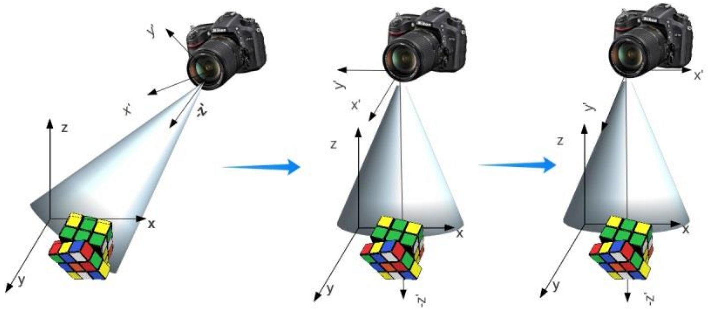
Size of Image: 11500*7500.

Triangles: 1 million

Calculation: 86.25 trillion

Shadow

- Graphic pipeline– analogous to taking a picture
  - Model transform: arrange objects in the scene to be photographed by the camera.
    - World coordinate system settled– RD
  - View transform: set up a camera on tripod pointing to the scene.
    - Transform the model to camera coordinate system.
      - Origin: camera position. Point to –Z axis
    - Translation and Rotation. $\boldsymbol{R} \times \boldsymbol{T}$



Place a Model → Model Transform → View Transform → Projection Transform → Viewport Transform → Display a 2D image

Shadow

- Projection transform

$$fovy = 2arctan(\frac{h}{2f})$$

Size of view in y $(tan(\frac{fovy}{2} * z), -tan(\frac{fovy}{2}) * z)$

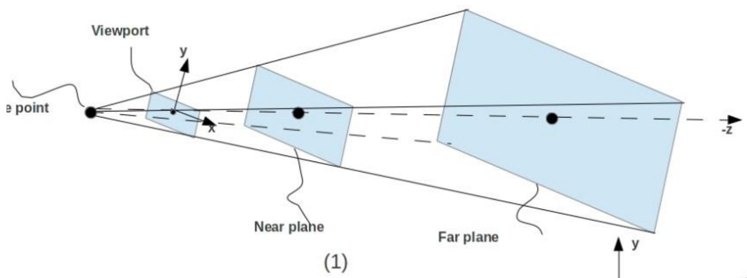
$$x_n = \frac{Y_c * F}{-Z_c * aspect}$$

Define F= $\frac{1}{tan(\frac{fovy}{2})}$

$$y_n = \frac{Y_c * F}{-Z_c}$$

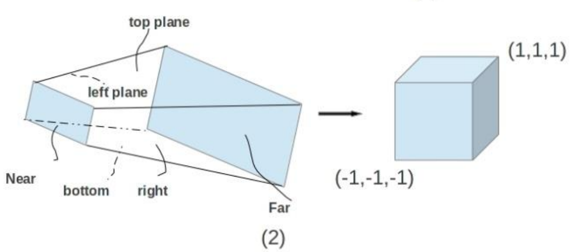
$$z_n = -1 + \frac{2(Z_c - Z_n)}{Z_f - Z_n}$$

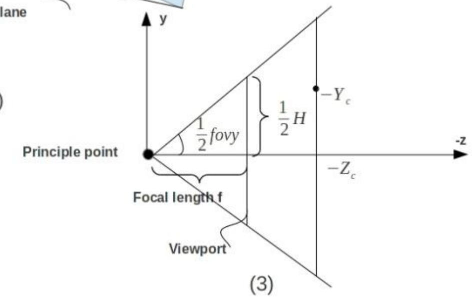
$$\begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{F}{aspect} & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & \frac{Z_f + Z_n}{Z_n - Z_f} & \frac{2Z_n Z_f}{Z_n - Z_f} \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$



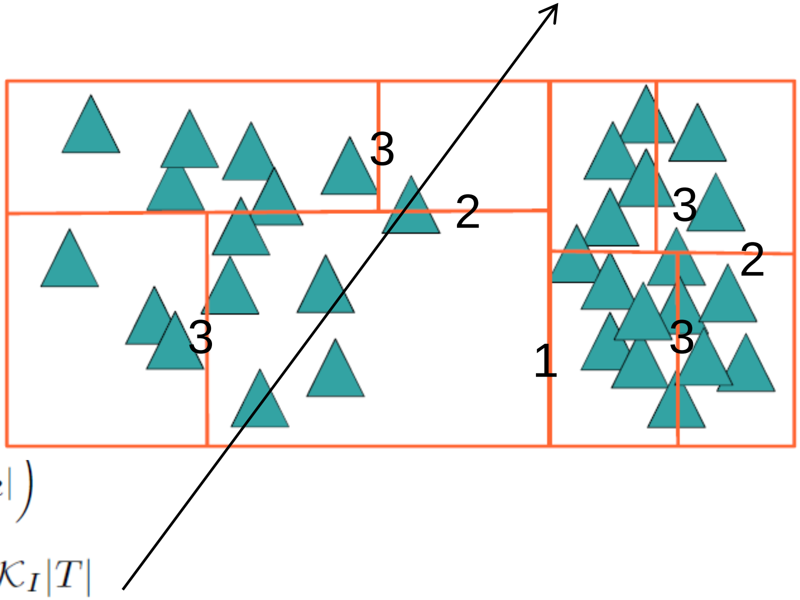
- View transform

$$\begin{bmatrix} \frac{w}{2} & 0 & 0 & \frac{w}{2} + x_p \\ 0 & \frac{h}{2} & 0 & \frac{h}{2} + y_p \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 1 \end{bmatrix}$$





Shadow

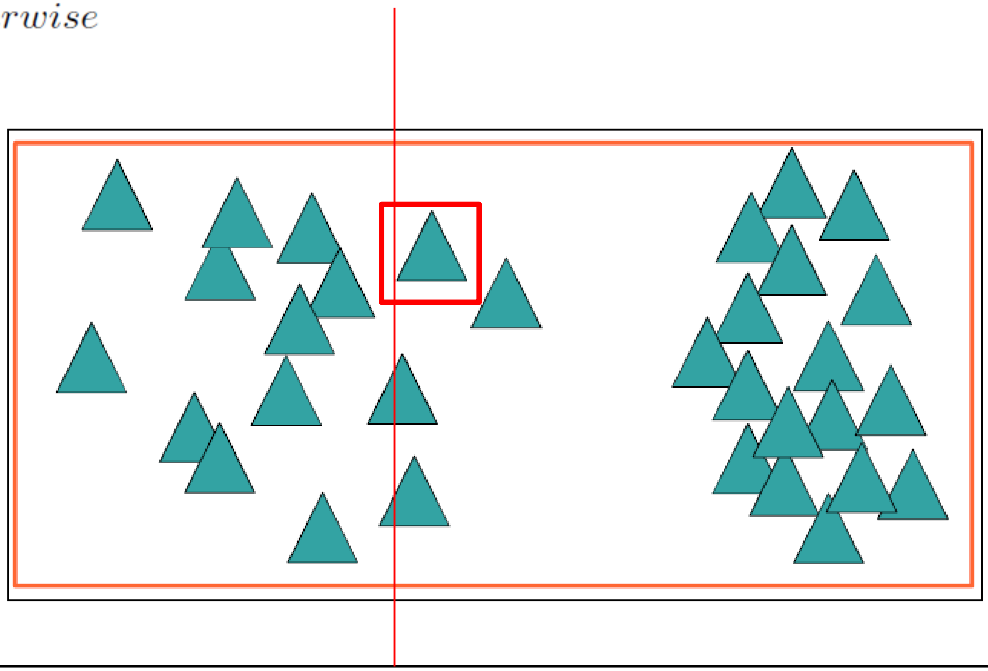


- Kd tree
  - Surface area heuristic (SAH)

$$\begin{aligned}\mathcal{C}_V(p) &\approx \mathcal{K}_T + \mathcal{P}_{[V_L|V]}|T_L|\mathcal{K}_I + \mathcal{P}_{[V_R|V]}|T_R|\mathcal{K}_I \\ &= \mathcal{K}_T + \mathcal{K}_I\left(\frac{\mathcal{SA}(V_L)}{\mathcal{SA}(V)}|T_L| + \frac{\mathcal{SA}(V_R)}{\mathcal{SA}(V)}|T_R|\right)\end{aligned}$$

$$Terminate(V,T) = \begin{cases} true & ; \min_p \mathcal{C}_V(p) > \mathcal{K}_I|T| \\ false & ; otherwise \end{cases}$$

- Naïve way for Kd tree construction
  - Find each optimal plane -- $O(N^2)$



$$\begin{aligned}T(N) &= N^2 + 2T(\frac{N}{2}) = \sum_{i=1}^{logN} 2^i\left(\frac{N}{2^i}\right)^2 \\ &= N^2\sum 2^{-i} \in O(N^2).\end{aligned}$$

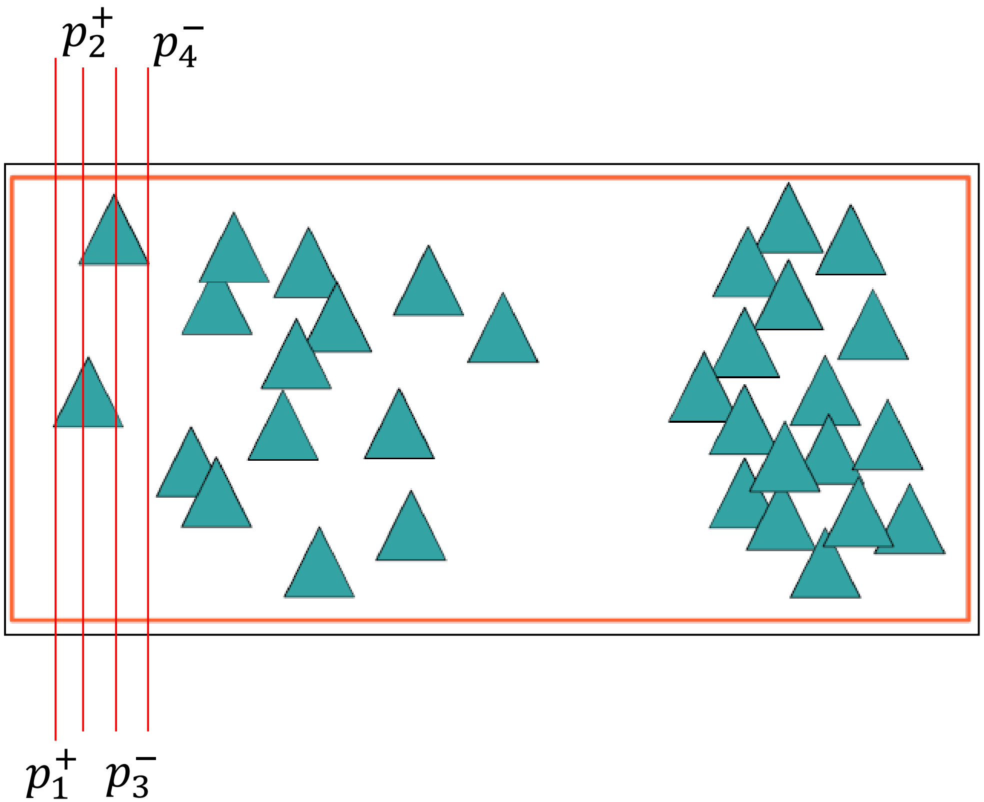
Shadow

- Kd tree

- Fast way- $O(Nlog^2N)$
  - Finding each optimal plane

  -- $O(NlogN)$
  - In total $O(Nlog^2N)$

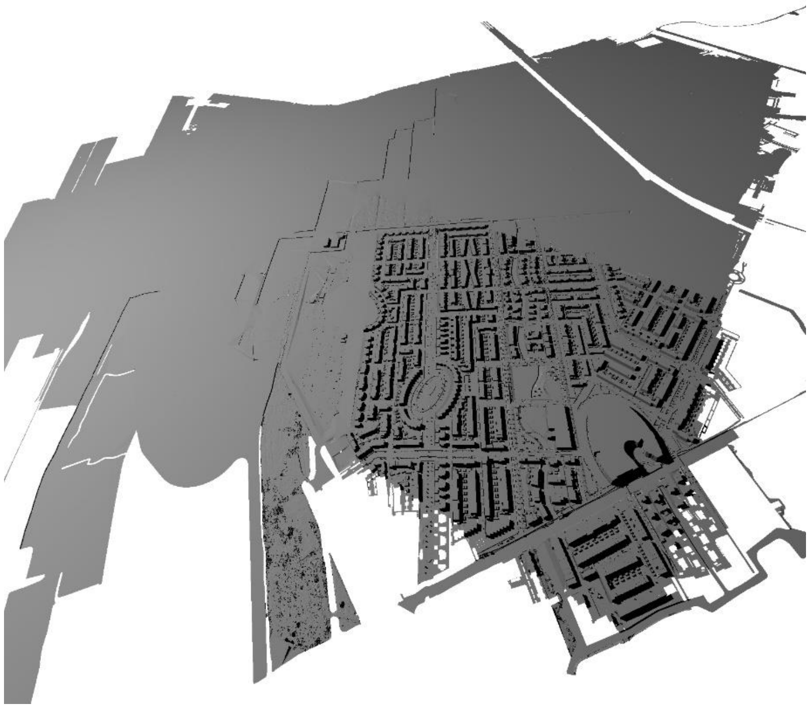
- Reduced calculation
  - 1 million triangles- 2500 times

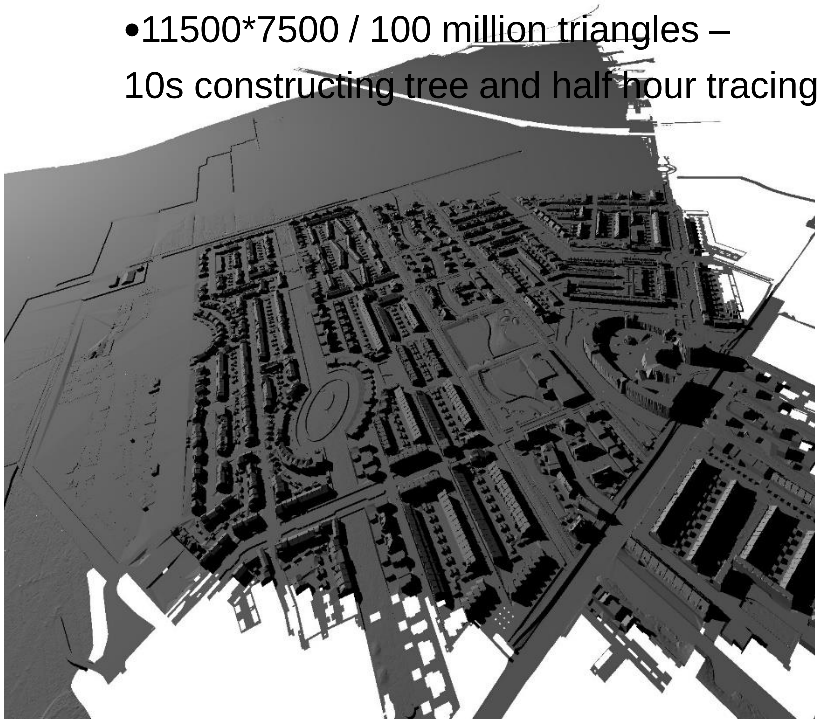


$$
\begin{aligned}
N_L^{(i)} &= N_L^{(i-1)} + p_{i-1}^{|} + p_{i-1}^{+} \\
N_R^{(i)} &= N_R^{(i-1)} - p_i^{|} - p_i^{-} \\
N_P^{(i)} &= p_{i-1}^{|}
\end{aligned}
$$

Shadow

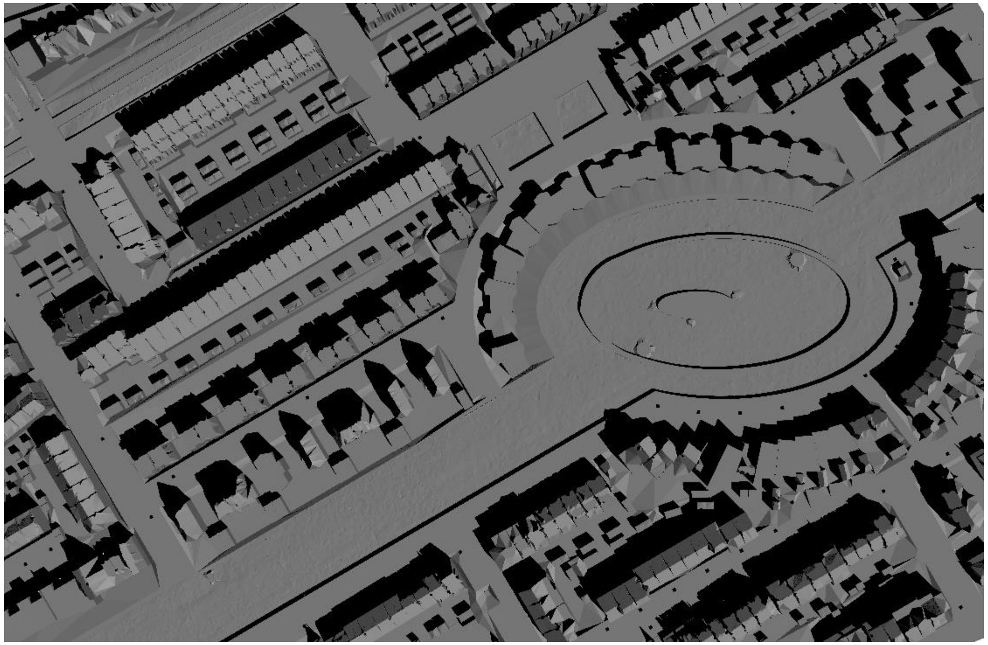
- Shadow
  - LoD1 model
  - LoD2 model
  - 11500*7500 / 100 million triangles – 10s constructing tree and half hour tracing





Shadow –To be done

- Shadow change
    - Accuracy of 3D model– assessment from original point cloud
    - Change in reality



- By using information in 3D model to help Shadow detection and change detection

Questions?