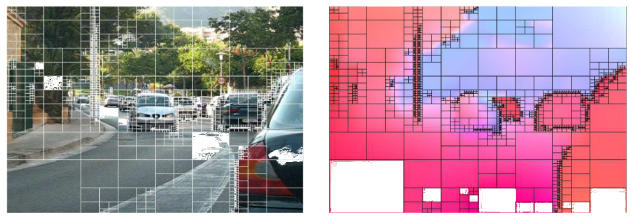
3D Machine Vision

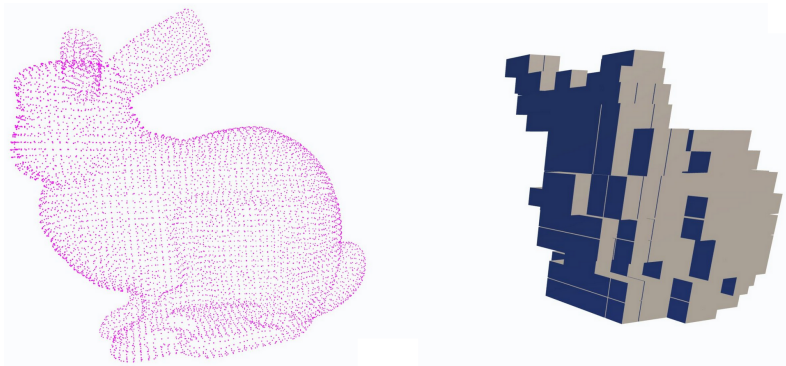
3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Representations of depth images

- Structured representations
- Unstructured representations
- Data formats for point clouds
- RGB-D images
- Voxel grids
- Quadtree and Octree
- kD-tree





3D Data Representations

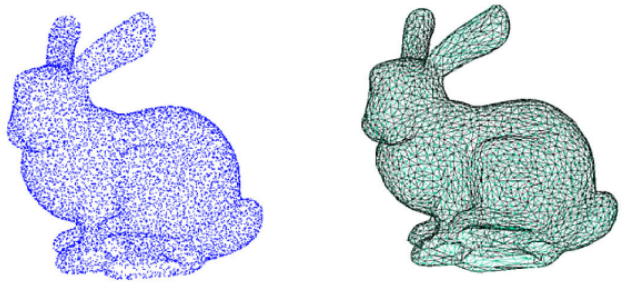
Representation of Depth Images

Unstructured Representations

- point clouds (list of 3D coordinates)
- polygonal grid (e.g. triangular grid)
- etc.

properties

- no fixed topology (arrangement in space).
- no uniform grid cell geometry
- flexible to use
- large memory requirement
- high computational effort for the computation of neighborhoods



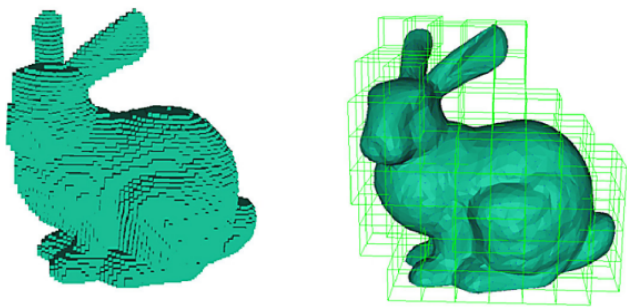
3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Representation of Depth Images

Structured representations

- RGB-D images (2D image + depth)
- Voxel grid (uniform 3D grid map)
- Octree (irregular 3D grid map)
- kD-tree
- etc.



properties

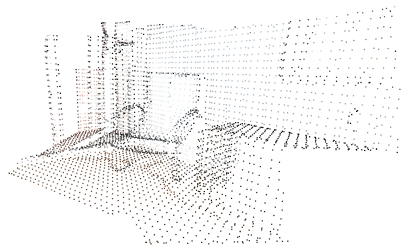
- regular topology
- not necessarily uniform grid cell geometry
- cells are present in a grid structur
- cells can be uniquely indexed by integer numbers
- low computational effort for the calculation of neighborhoods

3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

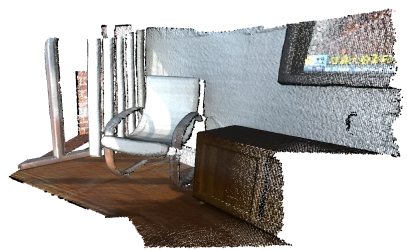
Point Clouds

A set of 3D coordinates is called a point cloud.
Further attributes can be assigned to the coordinates, e.g. color, intensity, normal, etc.



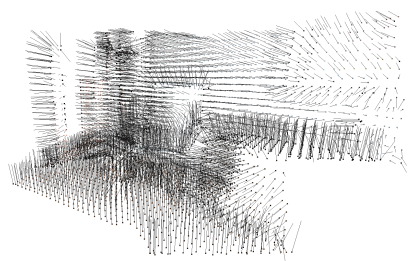
advantages of point clouds

- no discretization of the data
- the 3D area is not limited



disadvantages of point clouds

- no fixed topology $\rightarrow$ no direct representation of free or occupied space.
- neighborhood relations between points cannot be computed efficiently



3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Data formats for point clouds

There are different data formats for saving point clouds:

- OFF: Object File Format
- STL: Standard Triangulation/Tessellation Language
- PLY: Polygon File Format. Developed in the mid 90's by Greg Turk at Stanford University and published in 1994, hence also called Standford Triangle Format
- PCD: Point Cloud Data. Has some advantages over other formats. Can map the structure of RGB-D images and supports binary data types. This allows faster access to data $\rightarrow$ real-time processing. Also, n-D histograms can be stored as descriptors.

Greg Turk, Professor for Computer Graphics at Georgia Tech.

3D Data Representations

Data formats for point clouds

Use of data formats: Unambiguous and efficient description of individual points with attributes as lists of polygons:

- Attributes can be: coordinates, surface normals, color, transparency, texture coordinates, confidence values, 3D descriptors, etc.
- Different attributes possible for front and back side of a polygon
- Different data types possible: `char`, `short`, `int`, `float`, `double` etc.
- Storage of different file formats possible: text file (ASCII format) or binary file.

3D Data Representations

PLY: The Polygon File Format

The following elements can be defined in PLY format:

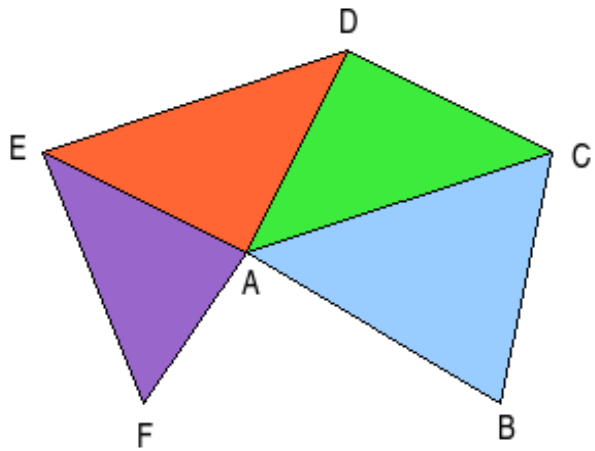
- `format`: File format (ascii, binary).
- `comment`: Comments describing the file content.
- `element vertex`: Number of objects, i.e. 3D points that make up the polygons of the surface (indexing starts at 0).
- `property`: properties of the objects with data type and name
- `element face`: Number of polygonal surface elements represented in the form of triangle fans.
- `property list`: A list encoding the surface elements.

3D Data Representations

Geometric Primitive: Triangel Fan

A triangle fan is a primitive in 3D computer graphics that saves memory and processing time. It describes a series of N connected triangles (triangle grid) of $N + 2$ points (nodes) sharing a central vertex and defining $N + 5$ triangle sides (edges).

- First element of the list encodes the number of polygon points
- Further elements of the list encode the index of the individual objects/coordinates
- Indexing of the objects starts at 0
- Second list element corresponds to the central corner point



Example: The list 4 0 1 2 3 describes a triangular fan of four points from the triangles 0,1,2 and 0,2,3.

3D Data Representations

PLY: The Polygon File Format

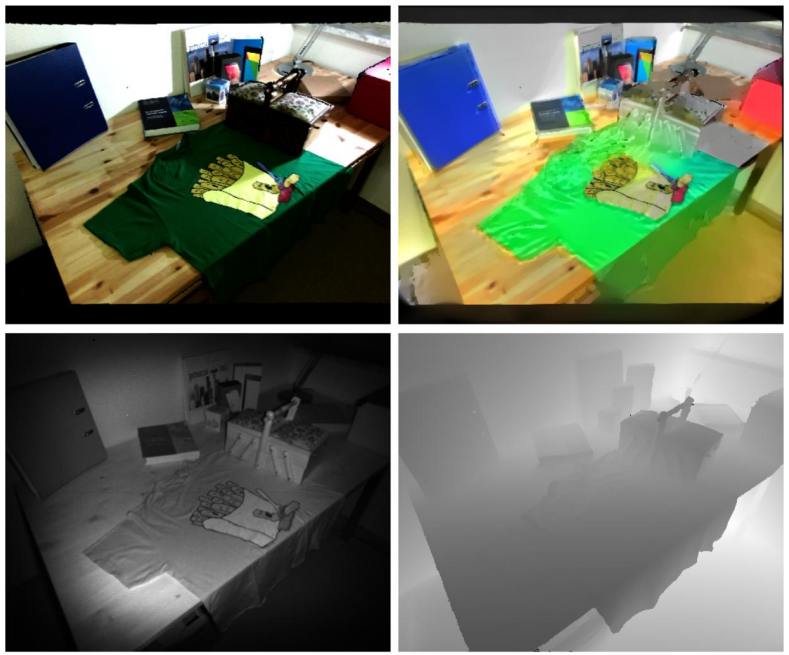
```
ply
format ascii 1.0                          { ascii/binary, format version number }
comment made by Greg Turk                 { comments keyword specified, like all lines }
comment
element vertex 8                          { define "vertex" element, 8 of them in file }
property float x                          { vertex contains float "x" coordinate }
property float y                          { y coordinate is also a vertex property }
property float z                          { z coordinate, too }
element face 6                            { there are 6 "face" elements in the file }
property list uchar int vertex_indices    { "vertex_indices" is a list of ints }
end_header                                { delimits the end of the header }
0 0 0                                     { start of vertex list }
0 0 1
0 1 1
0 1 0
1 0 0
1 0 1
1 1 1
1 1 0
4 0 1 2 3                                 { start of face list }
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
```

3D Data Representations

Depth Images and RGB-D Images

Technical University of Applied Sciences Würzburg-Schweinfurt

Depth images correspond to the measurement results of a 3D camera system. RGB-D images complement depth images with RGB color values.



Advantages

- depth values are arranged in a structured way in a 2D grid
- 2D neighborhoods can be quickly determined via two indices
- Each pixel of the image can be assigned a 3D coordinate, thus also a distance and a color value.
- Simple segmentation possible e.g. using quadtrees

Disadvantage

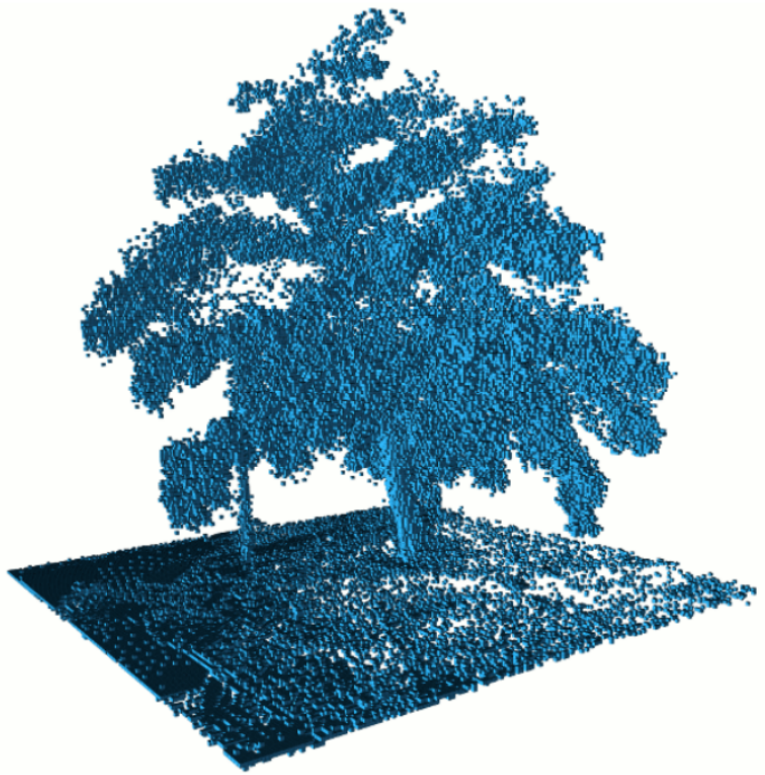
- Nearest 2D neighbor does not have to match nearest 3D neighbor

3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Voxel Grids - 3D Occupancy Grid Maps

Voxel grids overlay a 3D grid with a fixed voxel size over a 3D point cloud.



Advantages

- Volume representation
- Access via indexing in constant time
- Different states/values can be assigned to a voxel: occupied, free, probability, color.

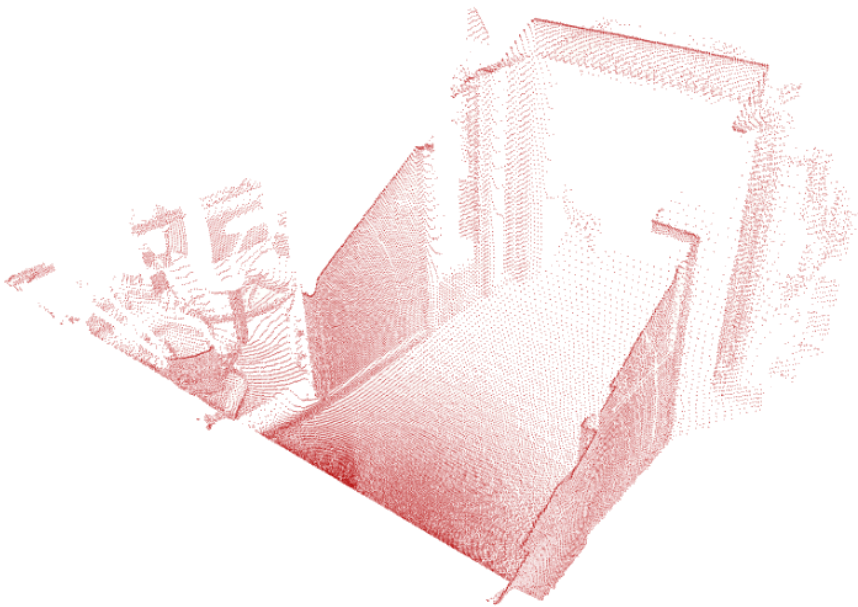
disadvantages

- limited range
- limited resolution
- discretization error

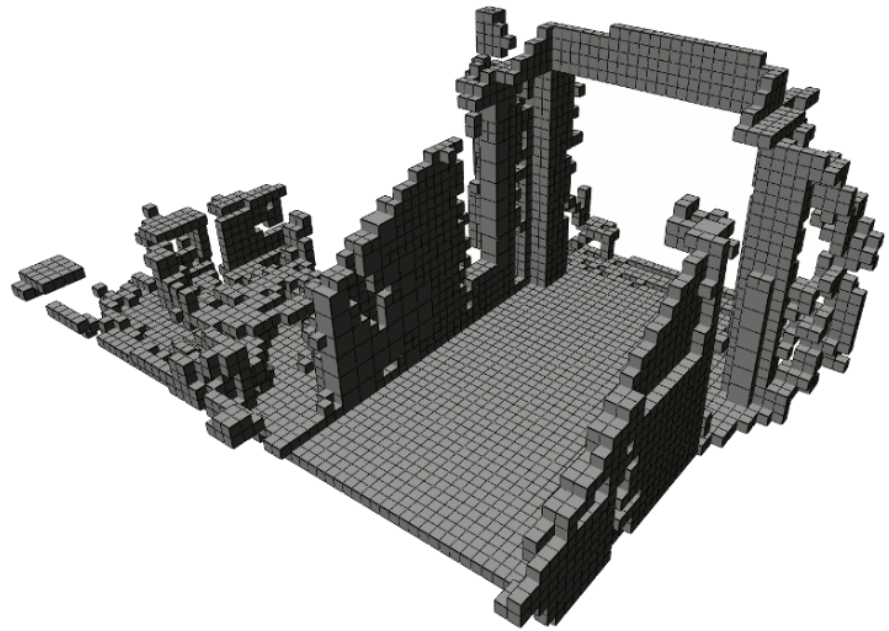
3D Data Representations

Voxel Grids - 3D Occupancy Grid Maps

Technical University of Applied Sciences Würzburg-Schweinfurt



Point Cloud



3D Occupancy Grid Map

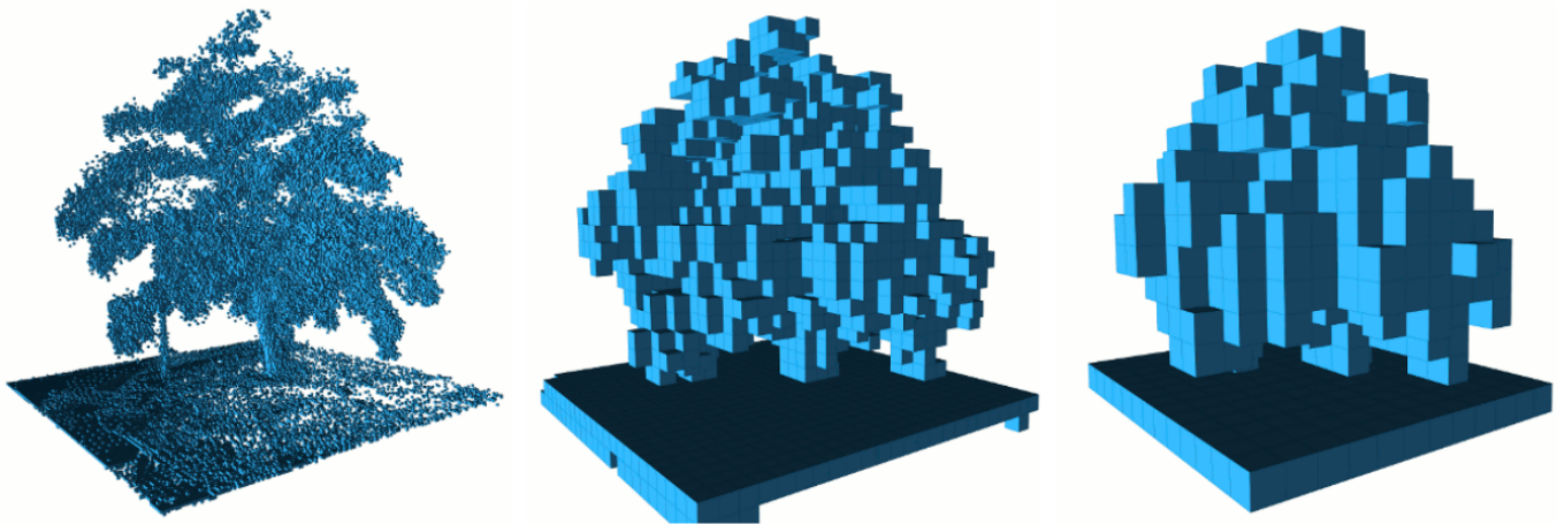
3D Data Representations

Voxel Grids - 3D Occupancy Grid Maps

Technical University of Applied Sciences Würzburg-Schweinfurt

Different levels of discretization:

- 0.08m
- 0.64n
- 1.28n



3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Voxel Grids - 3D Occupancy Grid Maps

Storing multiple attributes per voxel:

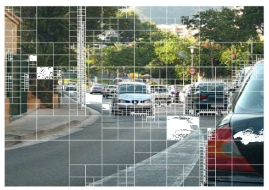
- Occupied or free
- and color

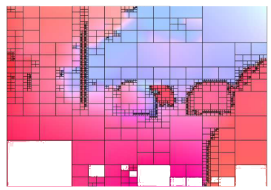
3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Quadtree

A quadtree can be used to recursively split a 2D space into cells of different sizes and store the result of the split in a tree structure.

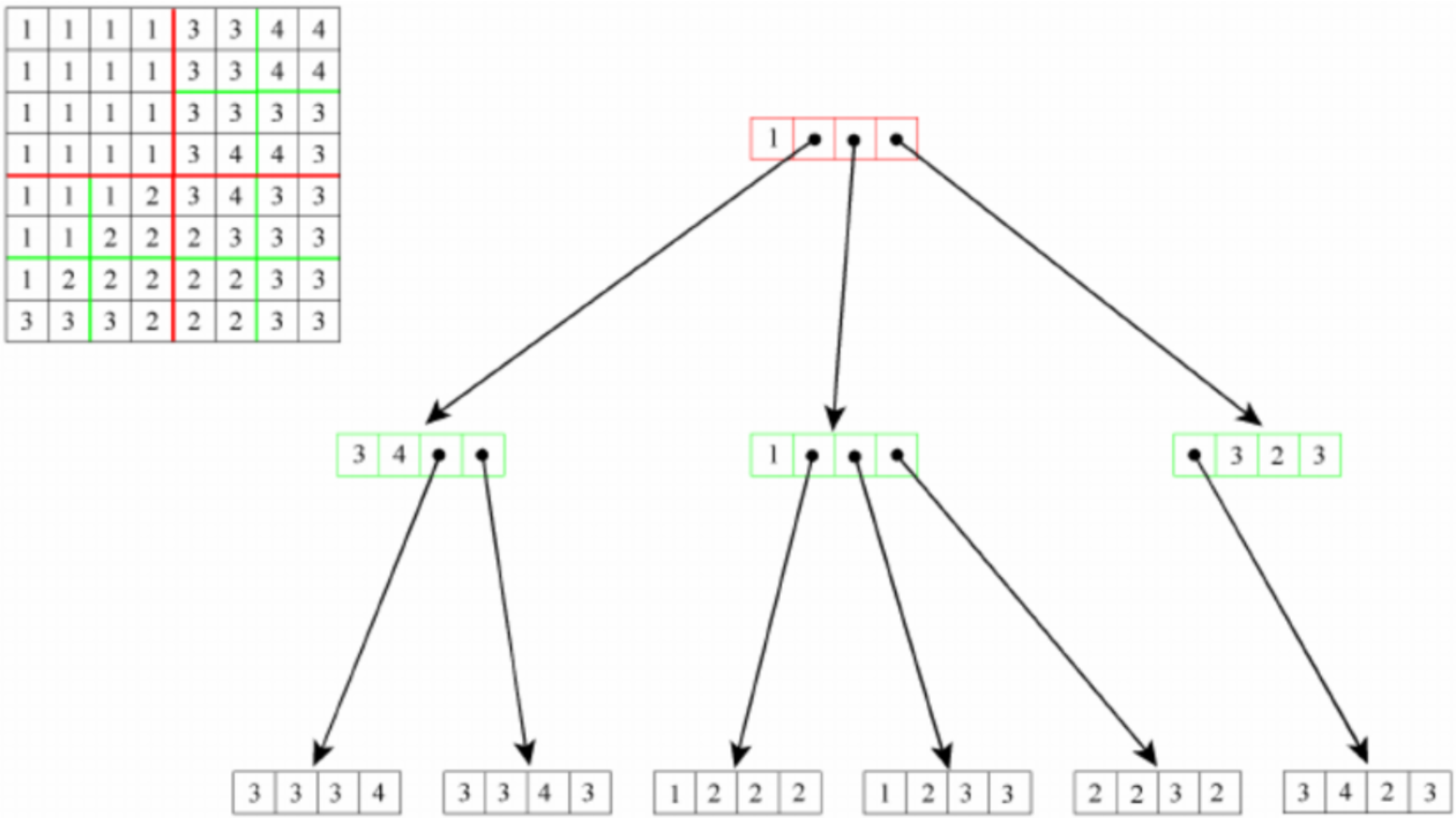




- Different criteria can be set when a cell is decomposed.
- The criterion corresponds to a comparison of the values in the cell with a model, e.g. comparison with the mean value and a threshold decision on the deviation.
- applications are the creation of 2D occupancy maps or the segmentation and compression of color and depth images.

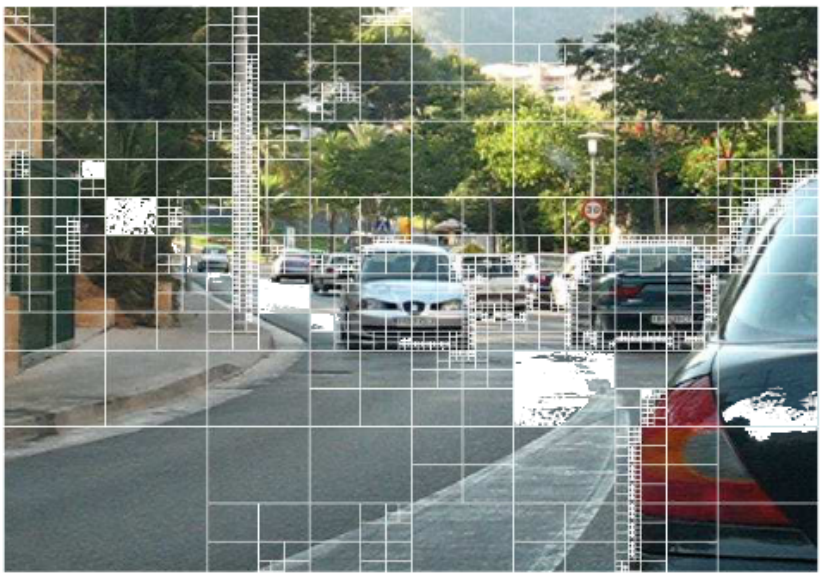
3D Data Representations

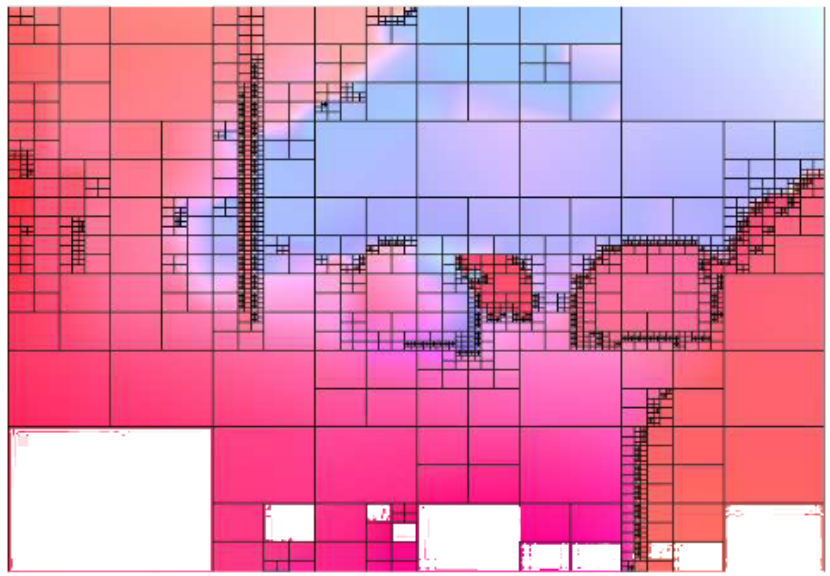
Quadtree - Splitting Principle



3D Data Representations

Quadtree - Example Motion Segmentation

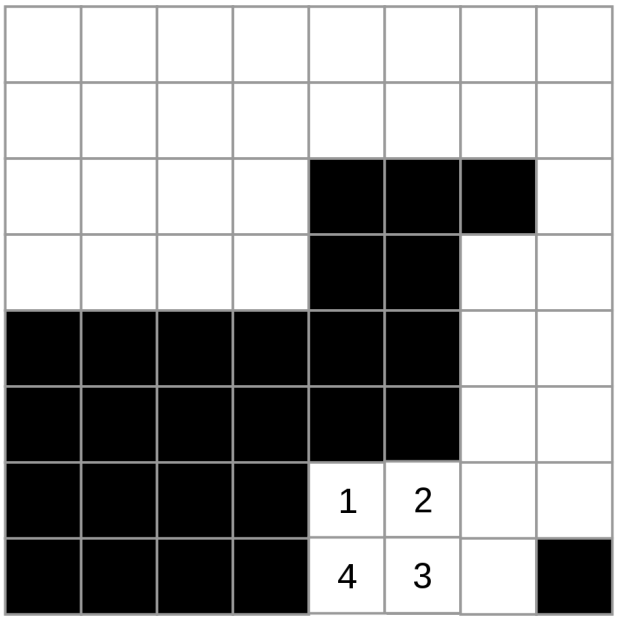


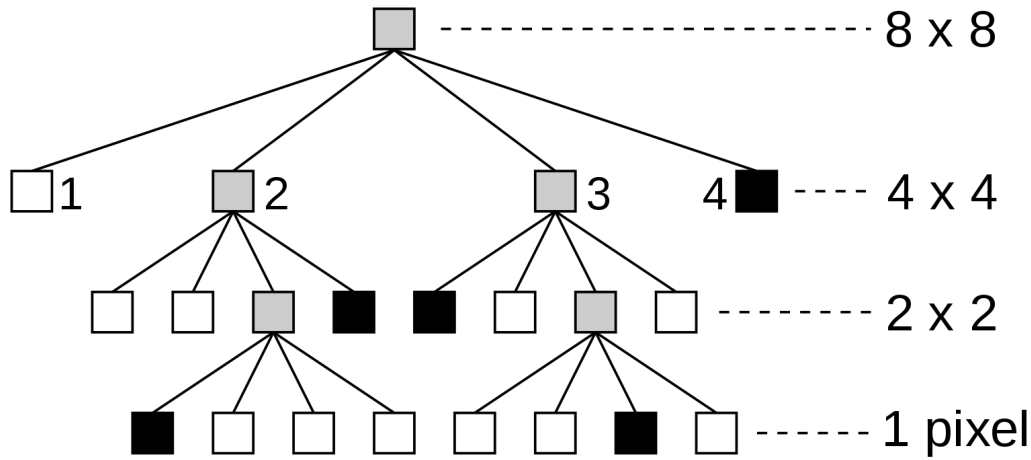


- Each block represents the optical flow via a polynomial model, e.g. affine model
- Split and Merge Segmentation: In addition a segmentation can be done based on merging similar blocks

3D Data Representations

Quadtree - Example 2D Occupancy Grid Map

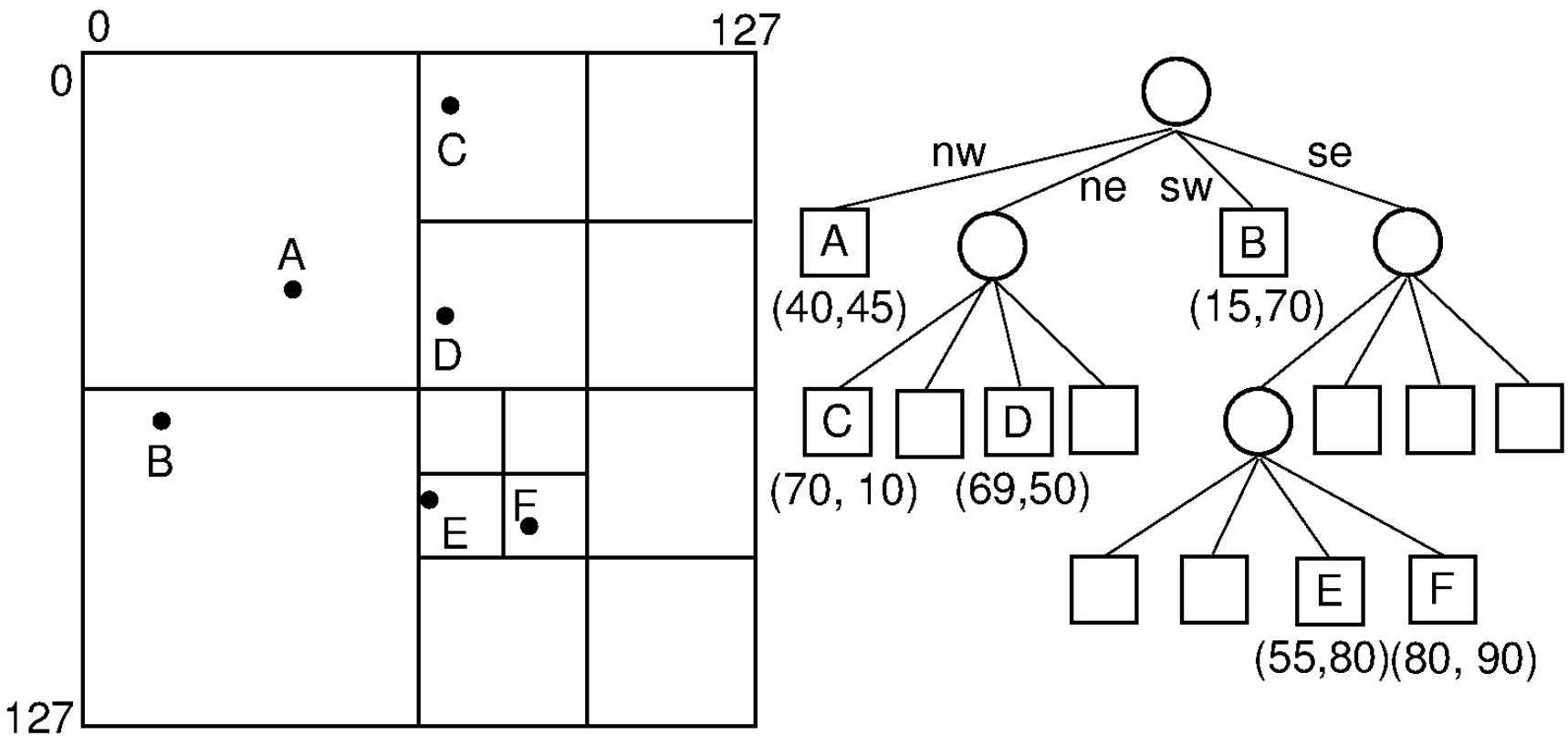




- Each block is either occupied (black) or free (white)

3D Data Representations

Quadtree - Example 2D Point-Region Quadtree



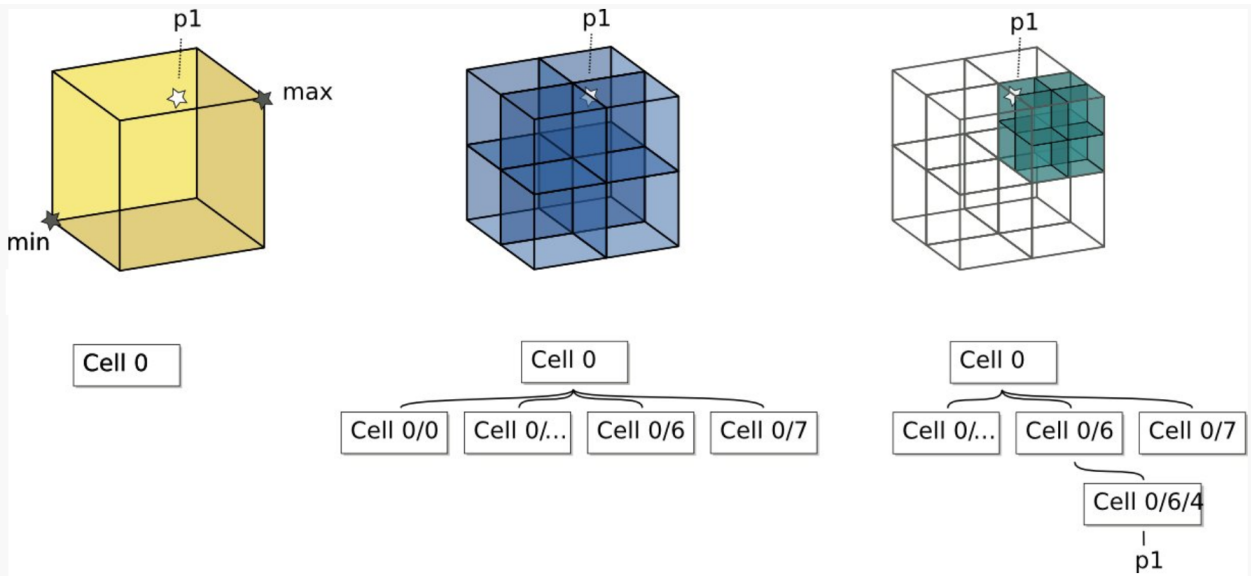
- At the end of splitting each block contains one data point of the point cloud

3D Data Representations

Octree

An octree can be used to recursively split a 3D space into voxels of different sizes and store the result of the split in a tree structure.

- effective representation of 3D occupancy maps
- as long as there is at least one 3D point in a cell and the finest discretization level is not yet reached, the cell will be further decomposed

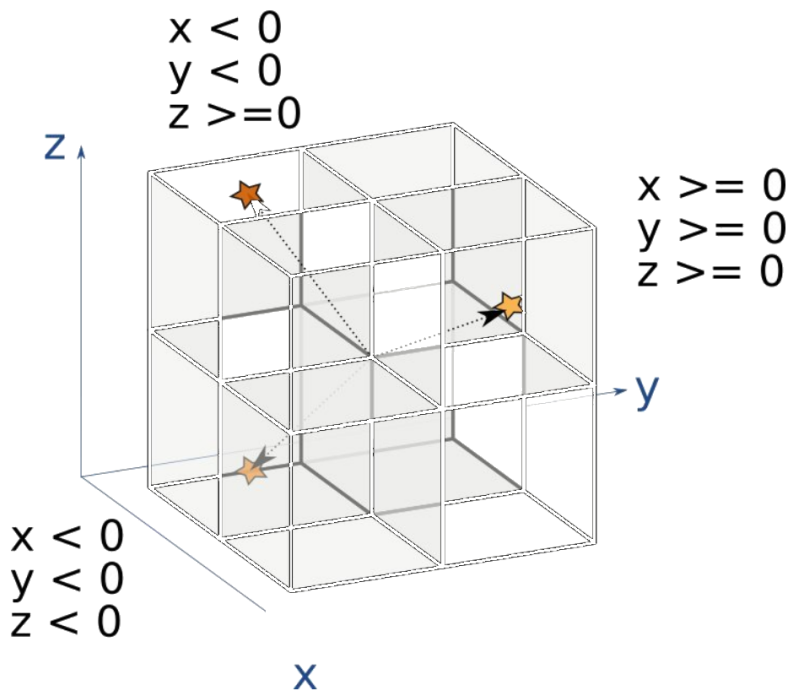


3D Data Representations

Octree

An octree can be used to recursively split a 3D space into voxels of different sizes and store the result of the split in a tree structure.

- For cell assignment the sign of the difference vector to the cell center point is sufficient
- Order of the cells is given



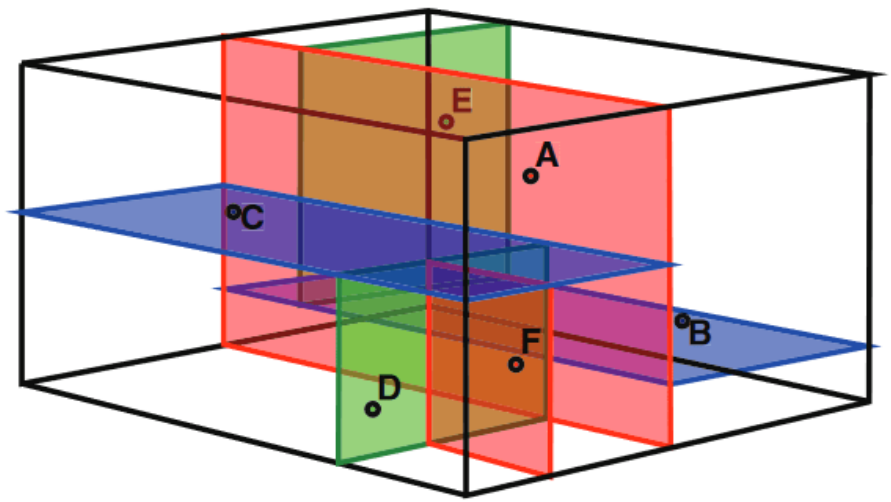
3D Data Representations

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

kD-tree

A k-dimensional tree or k-D-tree is a balanced search tree for storing points from the $\mathbb{R}^k$. Hyperplanes divide space recursively along the axes of the coordinate system.

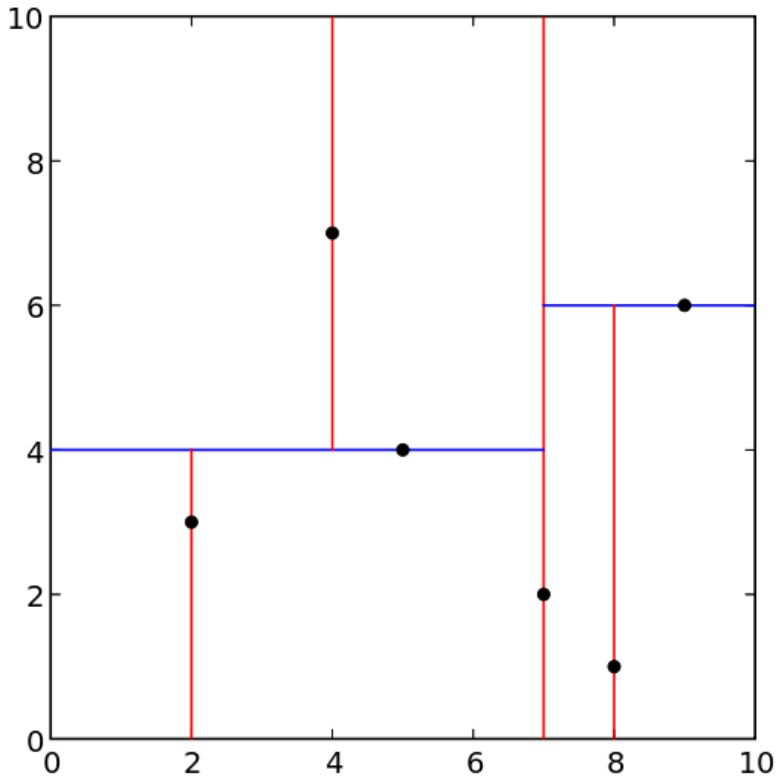
- Efficient search of k nearest neighbors (not the same k!)
- Search, insert, and delete points on average in logarithmic time possible
- Subdivision of the space depends on the coordinates of the points and is irregular
- the median per dimension defines the next hyperplane

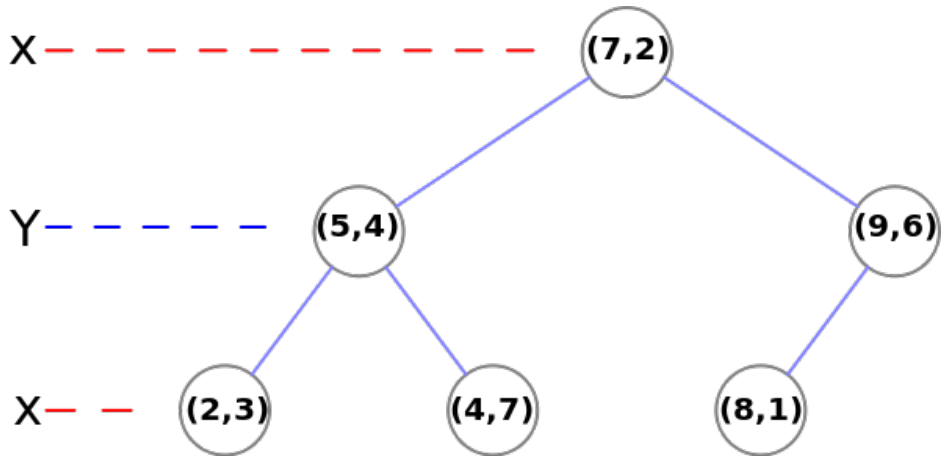


3D Data Representations

Example 2-D-tree

- Dimensions are traversed according to a fixed order, e.g. X→Y
- The median of each dimension divides the space recursively into two halves

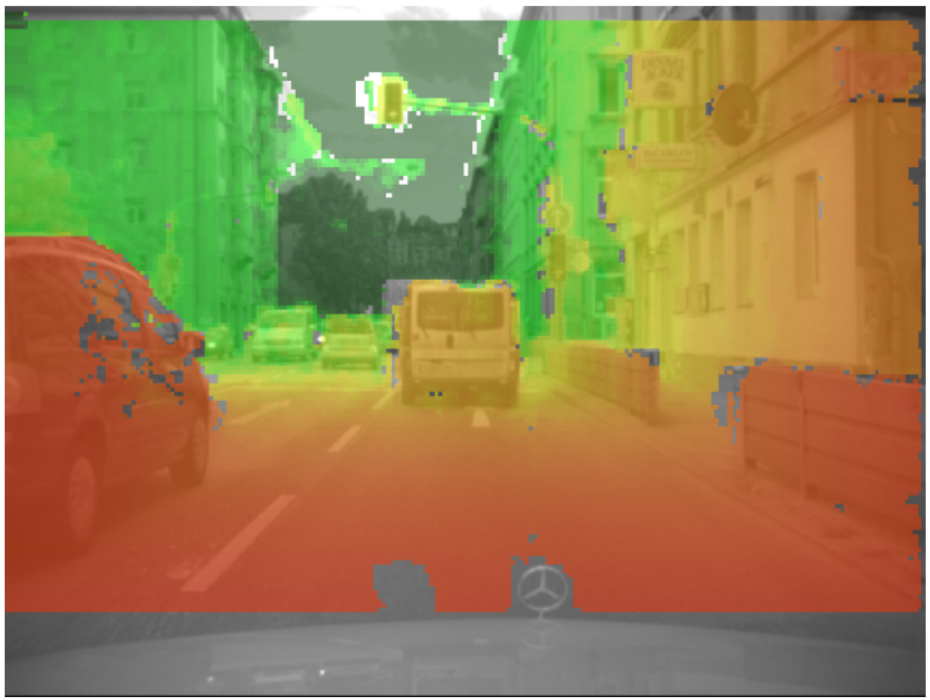




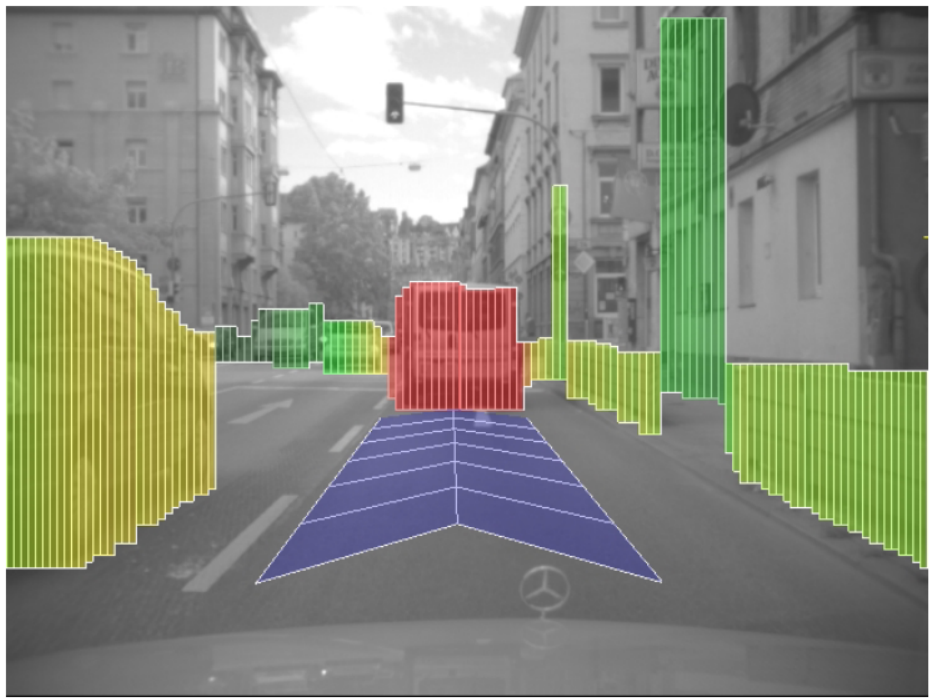
3D Data Representations

Hight Maps - Example: Stixels

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt



(a) Dense disparity image



(b) Stixel representation

- Mid-level representation: 1) find ground plane, 2) cluster objects based on stixels (columns of different height)