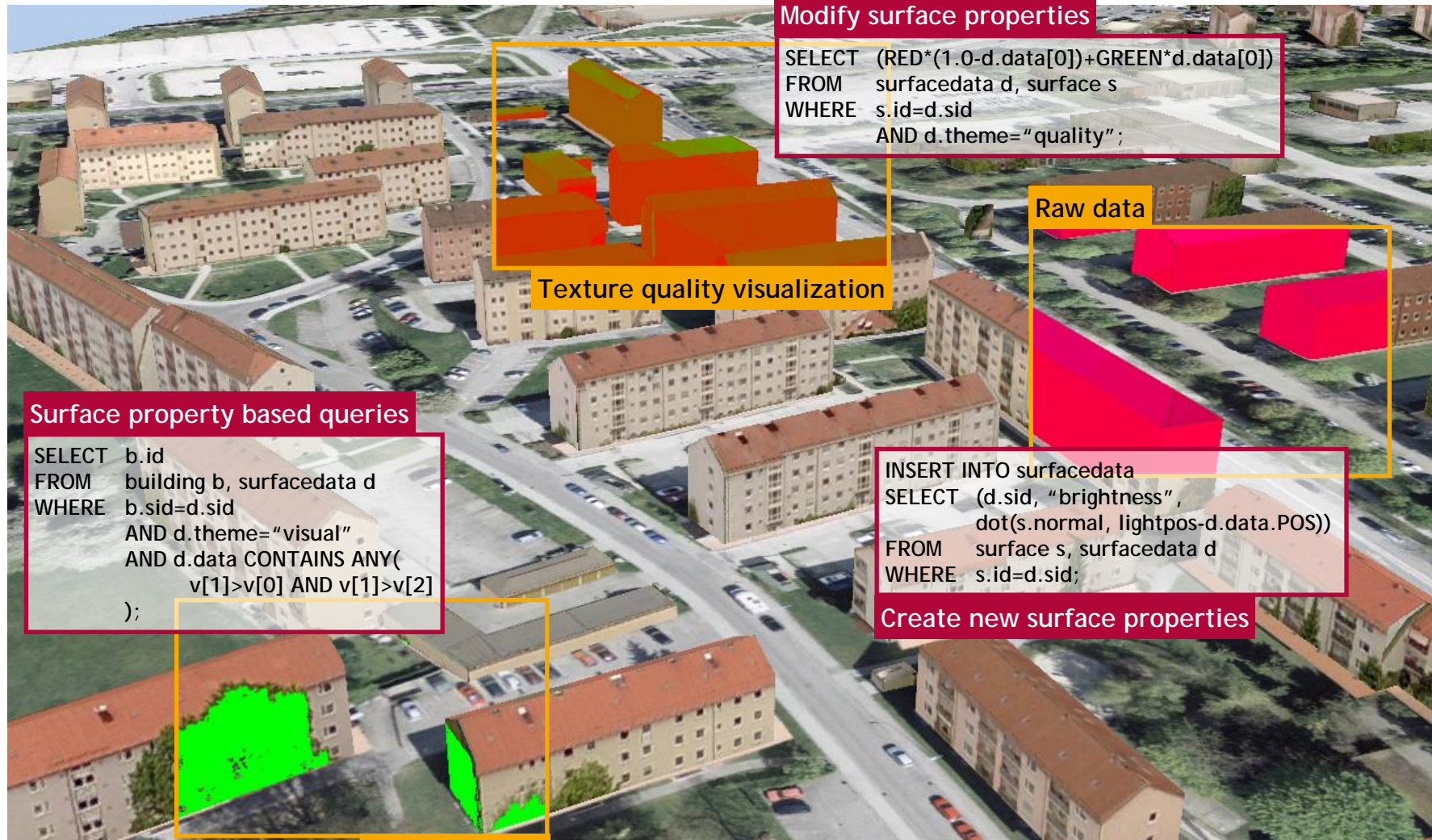


# Towards Integrating Feature Surface Properties into 3D GIS

Haik Lorenz and Jürgen Döllner

1



## Modify surface properties

```
SELECT (RED*(1.0-d.data[0])+GREEN*d.data[0])  
FROM surfacedata d, surface s  
WHERE s.id=d.sid  
AND d.theme="quality";
```

## Raw data

## Surface property based queries

```
SELECT b.id  
FROM building b, surfacedata d  
WHERE b.sid=d.sid  
AND d.theme="visual"  
AND d.data CONTAINS ANY(  
    v[1]>v[0] AND v[1]>v[2]  
);
```

```
INSERT INTO surfacedata  
SELECT (d.sid, "brightness",  
    dot(s.normal, lightpos-d.data.POS))  
FROM surface s, surfacedata d  
WHERE s.id=d.sid;
```

## Create new surface properties

## Texture quality visualization

## Highlight trees

# About Surface Properties

2

## Surface property properties:

- Surface properties are rasters relating to surfaces
  - Indirectly bound to features via feature surface
  - Mapping from raster location to 3D location
- A feature's surface property is a collection of 2D rasters
  - Demo dataset: 613 buildings, 10182 rasters, 250MB data

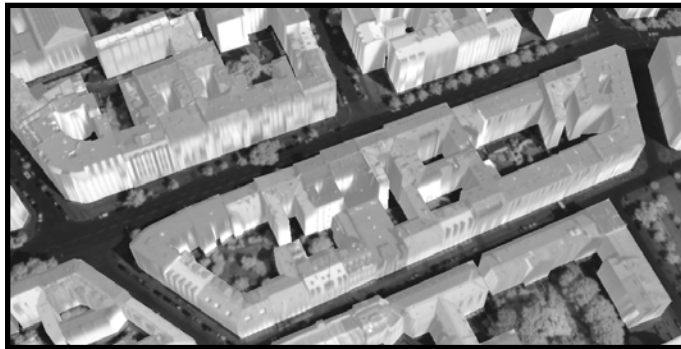
## Current situation:

- Automatic capturing possible and feasible
- Processing requires proprietary software
- Used mainly for visualization (surface properties = textures)
  - "Queries" by visual inspection

# Where to go - Examples

3

## 1. Identify poorly insulated buildings



```

SELECT  b.id
FROM    building b, surfacedata d
WHERE   b.sid=d.sid
        AND d.theme="infrared"
        AND d.data CONTAINS ANY(
                v[0]>20
        );
    
```

## 2. Wireless network coverage



[Image courtesy of  
Wavecall S.A., Switzerland.]

1. Create empty surface property
2. For each "datel"
  1. Create line of sight to each antenna
  2. Intersect lines with model geometry
3. Write max(signal strength) for all uninterrupted lines

## GIS component 1: Capturing

- Important operation: create empty surface property
- Existing examples: texture atlas creation, surface parameterization
  - E.g.: Praun et al.: “Lapped Textures”, 2000.

## GIS component 2: Storage

- Deals with massive image patch collections
- Add 3D spatial indices
- Existing examples: texture streaming approaches
  - E.g.: Buchholz and Döllner: “View-Dependent Rendering of Multiresolution Texture-Atlases”, 2005.

## GIS component 3: Retrieval

- Implement operations on/between surface property data
  - Per-"date" operations
  - Statistics/aggregation operations over surface (patches)
  - Location-based operations
- Requires tight link between features, surfaces, and rasters
- Existing examples: image operations in DBMS, fragment shaders in Computer Graphics

## GIS component 4: Analysis

- Requires writing/updating surface property data
  - Complex per-"date1" functions
  - Includes 3D location of a "date1"
- Similar to fragment shaders in Computer Graphics

## GIS component 5: Portrayal

- Requires definition of surface property usage/handling
- Can rely on Retrieval/Analysis functionality
- Existing examples: Collada (Computer Graphics data exchange format)
  - Khronos Group Inc.: "COLLADA - Digital Asset Schema Release 1.4.1 - Specification", 2006.

# The End

7



## Thank you for your attention!

Contact:

[haik.lorenz@hpi.uni-potsdam.de](mailto:haik.lorenz@hpi.uni-potsdam.de)

Research group "3D Geoinformation": [www.3dgi.de](http://www.3dgi.de)

[www.hpi3d.de](http://www.hpi3d.de)

Hasso-Plattner-Institute, University of Potsdam, Germany

supported by:



03/04/2009