AUTOMATED SELECTION OF TERRESTRIAL IMAGES FROM SEQUENCES FOR THE TEXTURE MAPPING OF 3D CITY MODELS

Sébastien Bénitez and Caroline Baillard
SIRADEL, 3 allée Adolphe Bobierre CS 24343, 35043 Rennes, France
sbenitez@siradel.com

KEY WORDS: Building, Texture, Image, Sequences, Terrestrial, Automation.

ABSTRACT:
The final purpose of this study is to texture map existing 3D building models using calibrated images acquired with a terrestrial vehicle. This paper focuses on the preliminary step of automated selection of texture images from a sequence. Although not particularly complex, this step is particularly important for large-scale facade mapping where thousands of images might be available. Three methods inspired from well-know computer graphics techniques are compared: one is 2D-based and relies on the analysis of a 2D map; the two other methods use the information provided by a 3D vector database describing buildings. The 2D approach is satisfactory in most cases, but facades located behind low buildings cannot be textured. The 3D approaches provide more exhaustive wall textures. In particular, a wall-by-wall analysis based on 3D ray tracing is a good compromise to achieve a relevant selection whilst limiting computation.

1. INTRODUCTION

With the development of faster computers and more accurate sensors (cameras and lasers), the automatic and large-scale production of a virtual 3D world very close to ground truth has become realistic. Several research laboratories around the world have been working on this issue for some years. Früh and Zakhor have proposed a method for automatically producing 3D city models using a land-based mobile mapping system equipped with lasers and cameras; the laser points are registered with an existing Digital Elevation Model or vector map, then merged with aerial LIDAR data (Früh and Zakhor, 2003; Früh and Zakhor, 2004). At the French National Geographical Institute (IGN), the mobile mapping system Stereopolis has been designed for capturing various kinds of information in urban areas, including laser points and texture images of building facades (Bentrah *et al*., 2004). The CAOR laboratory from ENSMP has also been working on a mobile system named LARA-3D for the acquisition of 3D models in urban areas (Brun *et al.,* 2007; Goulette *et al.,* 2007), based on laser point clouds, a fish-eye camera, and possibly an external Digital Elevation Model. Recently, a number of private companies have commercialized their own mobile mapping systems for 3D city modeling, like StreetMapper or TITAN for instance (Hunter, 2009; Mrstik *et al*., 2009). The purpose of such systems is often the 3D modeling as well as the texture mapping of the 3D models.
In this study we are interested in texturing existing 3D building models by mapping terrestrial images onto the provided façade planes. As a part of the mapping strategy, one first needs to determine which images each façade can be seen from. It is particularly important for large-scale facade texture mapping where thousands of images can be available. Every single image can be relevant for the final texturing stage. There are few references on this issue. In (Pénard *et al.,* 2005) a 2D map is used to extract the main building facades and the corresponding images. All the images viewing at least a part of a façade are selected. In (Haala, 2004), a panoramic camera is used and a single image is sufficient to provide texture for many façades. Given a façade, the best view is the one providing the highest resolution. It is selected by analyzing the orientations and distances of the building facades in relation to the camera stations. In (Allène, 2008), a façade is represented by a mesh. Each face of the mesh is associated to one input view by minimizing an energy function combining the total number of texels representing the mesh in the images, and the color continuity between two neighbouring faces.
In our study, only two triangles per facade are available, and a façade texture generally consists of a mixture of 4 to 12 input views. The following mapping strategy has been chosen for texturing a given façade:

- Pre-selecting a set of relevant input images, from which the façade can be seen;
- Merging these images into a single texture image;
- Registering the texture image with the existing façade 3D model.

This paper only focuses on the first stage. The purpose of this operation is to select a set of potentially useful images based on purely geometrical criteria. The generation of a seamless texture image without occlusion artifacts will be handled within the second stage. Three possible approaches for the image pre-selection are presented and discussed. The first approach is similar to the one used in (Pénard *et al.,* 2005) and relies on the analysis of a 2D map. The two other methods use the information provided by a 3D vector database describing buildings. All methods are based on standard techniques commonly used in computer graphics for visibility computations, namely the ray-tracing and z-buffering techniques (Strasser, 1974). These two techniques have now been used for decades and are very well known in the computer graphics community. They can easily be optimized and accelerated via a hardware implementation.
This paper is organized as follows. Section 2 presents the test data set used for this study. Sections 3, 4 and 5 detail the three selection methods. The results and perspectives are discussed in section 6.

2. TEST DATASET

The test area is a part of the historical center of the city of Rennes in France. It is 1 km² wide and corresponds to the densest part of the city. Existing 3D building models were provided with an absolute accuracy around 1m. It contains 1475 buildings consisting of 11408 walls. The texture image database associated to the area was simulated via a virtual path created through the streets. A point was created every 5 meters along this path. Each point is associated to two cameras facing the left and the right sides of the path. The camera centers are located at 2.3 meters above the ground in order to simulate a vehicle height. The internal and external parameters of the cameras are approximately known. The path is about 4.9 kilometers long, and it contains 990 points and 1980 cameras views (see Figure 1). It includes loops, self-intersections and close parallel roads. As a result a building wall can be seen from several locations within the path.

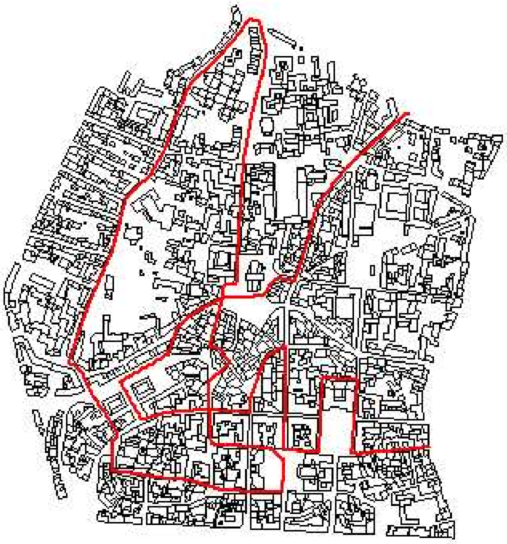


Figure 1. Test area, Rennes historical center. The virtual path is depicted in red.

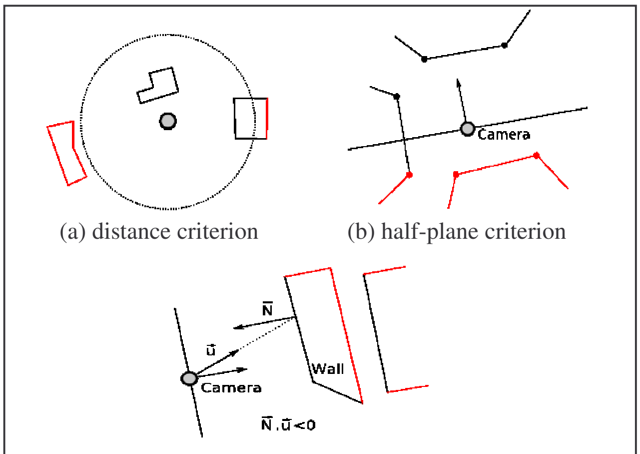
3. 2D RAY TRACING

3.1 Principle

The 2D approach is based on ray-tracing. Each camera is analyzed in turn. The walls are represented by 2D segments. For each camera a set of compatible wall segments is pre-selected using three criteria (see Figure 2):

- Distance criterion: the wall is located within a given distance from the camera center.
- Half-plane criterion: at least one point of the wall segment is located in the half-space in front of the camera
- Backface culling criterion: the wall is facing the camera.

The compatible wall segments define a set of candidate walls that might be visible from the current camera. An example of pre-selection is shown in Figure 10a-b.



(a) distance criterion (b) half-plane criterion

(c) backface culling criterion

Figure 2. The three criteria for the selection of candidate walls: (black=pre-selected walls, red=rejected walls)

The 2D-tracing technique is then applied to the candidate wall segments. First a beam of 2D lines is defined passing through the camera center point and regularly distributed within the field of view of the camera. Then the closest intersected candidate wall segment is selected as a visible wall. When all the cameras have been processed then each wall can be associated to the list of cameras that can view it.

3.2 Test results

The method was tested with various numbers of rays per camera. The distance threshold was arbitrarily set to 150m, distance above which the texture resolution is low enough to be discarded. The computing time includes reading and exporting steps. Numerical results are shown in Table 1. Between 10 and 13% of the walls are detected as visible by the process. Figure 3 shows the evolution of the wall number and the computing time with the number of rays. A qualitative example of selected walls can be found in Figure 10c.

Ray #	Total # of selected walls	Avg # of cameras per wall	Computing time
10	1176 (10.3%)	4.54	7s
50	1391 (12.1%)	4.95	11s
100	1450 (12.7%)	5.04	17s
500	1507 (13.2%)	5.14	50s

Table 1. Results of 2D ray tracing

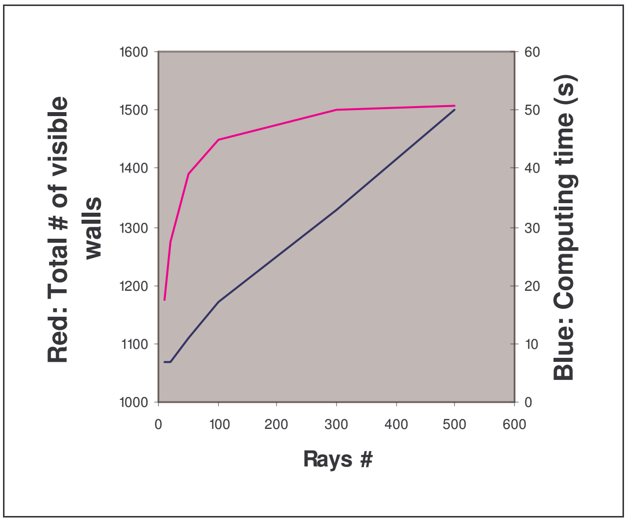
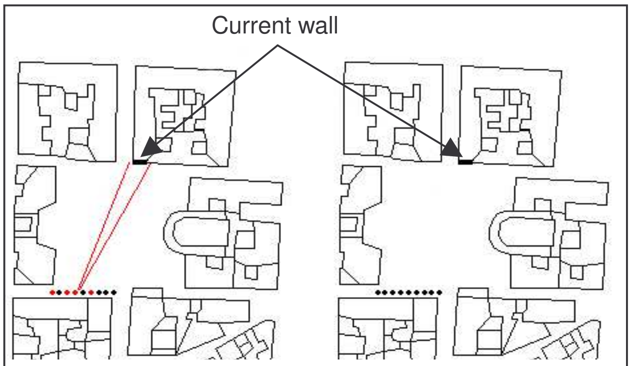


Figure 3 –Number of visible walls and computing time in relation to the number of rays

3.3 Discussion

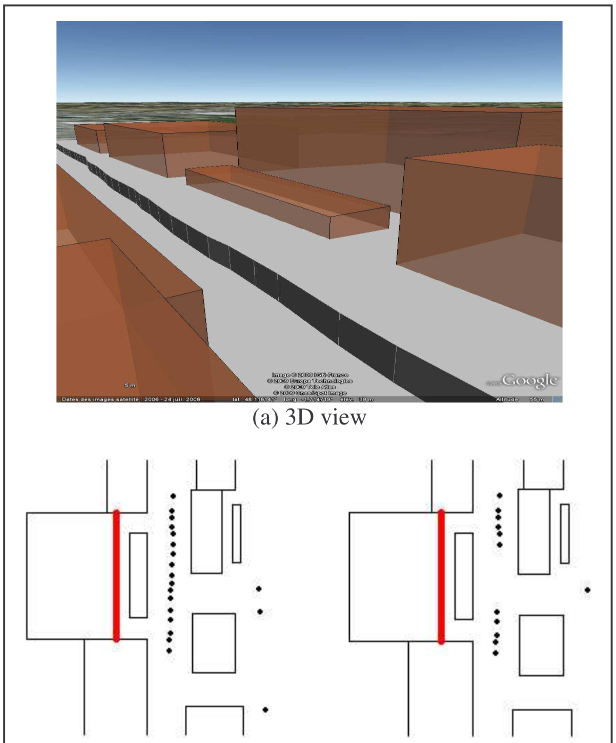
The variations in the number of selected walls come either from walls located far away of the camera, or from walls almost aligned with the camera center. When the number of rays is small, then many walls are located between two rays and are therefore not selected (see Figure 4). In our configuration, a number of rays around 100 seems to be a good compromise to get a maximum number of relevant images per building wall.

The main advantage of the 2D approach is the speed. It is also very simple and quick to implement. However it does not take building heights into account. Yet a low building (garage, shop, etc) may only mask the bottom part of higher buildings located behind it, especially if the camera is located on the top of a vehicle (see examples in Figure 5 and Figure 11a-c). Therefore it seems very important to make use of 3D information within the selection process.



(a) 10 rays per camera (b) 50 rays per camera

Figure 4. Example of a “missed” wall. (a) 10 rays per camera: the current wall is “missed” by several cameras (red dots); (b) 50 rays per camera: the current wall is seen by all the cameras (black dots)



(a) 3D view

(b) Cameras seeing the red wall (c) Cameras selected with the 2D approach

Figure 5. Example of incomplete camera selection

4. 3D Z-BUFFERING

4.1 Principle

The second approach is 3D-based and relies on a z-buffer technique. Each camera is analysed in turn. A set of candidate walls is first associated to the current camera as in described in section 3, using distance, half-plane and backface culling criteria. The camera is then associated to a label image identifying the walls seen by the camera, and a depth image indicating the distance from the camera centre to the walls. Finally, after all the cameras have been processed, each wall can be associated to the list of cameras that can view it.

4.2 Test results

In order to reduce computing time, the distance and label images are sub-sampled at coarser resolutions. The tests were performed at a sampling resolution of 5, 10 and 20 pixels. They are shown in Table 2. It is important to note that the algorithm was not optimised and the graphical card not used. An example of depth image is shown in Figure 6.

Z-buffer resolution	Image size (hxw)	Total # of visible walls	Avg # of cam. per wall	Computing time
5 pixels	216x384	2310 (20.2%)	4.40	61min58s
10 pixels	108x192	2249 (19.7%)	4.41	52min36s
20 pixels	54x96	2186 (19.2%)	4.35	43min54s

Table 2. Results of 3D ray tracing

Figure 6. Example of a depth image

4.3 Discussion

Using this approach, 50% more walls can be textured. In particular, all facades located behind other buildings can now be textured, whereas they were discarded with the 2D approach. In the example of Figure 10d, the circled area shows an example of a high building visible from the current camera but only selected with the 3D approach. As the measures are very dense, even small walls, walls distant from the path or wall aligned with the path can theoretically be textured.

In return, many selected walls are only partly visible, and would actually have a very poor texture quality. It is important to introduce a contribution culling technique, in order to discard wall images inappropriate for texture mapping. In the current implementation, another drawback of the method is the computing time. Using a hardware implementation directly into the Graphical Processing Unit of the graphic card should solve this problem. A hierarchical z-buffer technique could also be investigated (Greene, 1993). Finally, the selection process must be entirely completed before being able to further process a façade, which might not be compatible with a large-scale production process.

5. 3D RAY TRACING

5.1 Principle

The last approach combines the main advantages of the two previous ones: speed and use of 3D information. It is a 3D extension of the 2D approach based on ray tracing. However, the analysis is performed wall-by-wall rather than camera-by-camera. Given a wall, a set of candidate cameras is selected using a method similar to the one described in section 3:

- The camera is located within a given distance from the wall (distance measured at closest point).
- The camera center point is located in the half-space in front of the wall.
- The camera is facing the wall plane.

In order to reduce computing time and improve texture quality, an additional criterion has been introduced: cameras that are almost aligned with the wall plane are discarded. A maximum threshold on the angle defined by the wall plane and the camera directions is introduced (see Figure 7). This filtering step is an extension of the backface culling criterion.

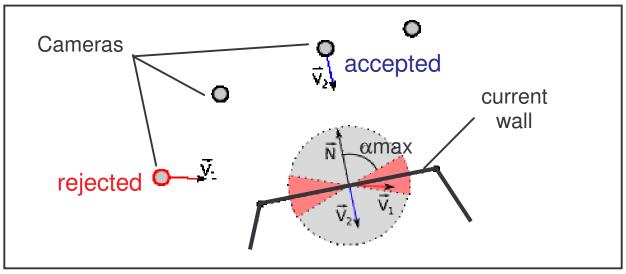


Figure 7. Angle criterion for the pre-selection of candidate cameras: the cameras with their direction vector in the red angular area are discarded

For each candidate camera a grid on the camera plane is defined. Each grid point defines a 3D ray passing through the camera center point. The 3D rays not intersecting the current wall are ignored. The remaining 3D rays are tested with respect to all the walls compatible with the camera (pre-selection method described in section 3): any ray intersecting a wall face closer than the current one is discarded. The candidate camera is finally selected as viewing the current wall, if at least one of the rays has not been discarded. The process is illustrated in Figure 8.

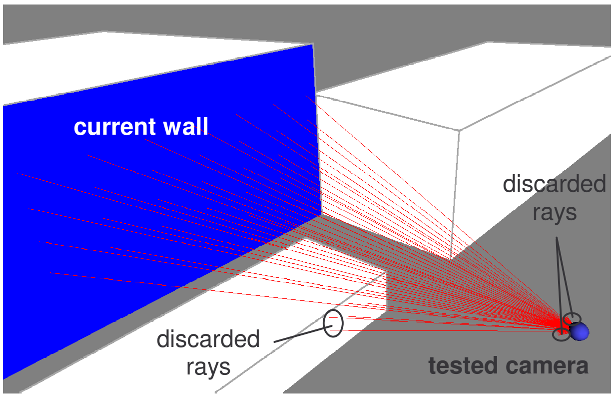


Figure 8. Principle of 3D ray tracing: the rays launched from the tested camera are discarded if they do not intersect the current wall (see rays on the extreme sides) or if they first intersect a closer wall (see rays on the left)

5.2 Test results

The method was tested with 10x10 and 20x20 rays per camera, with and without threshold on the angle during pre-selection. The threshold on angle was set to $\frac{3\pi}{8}$ radians when applied. The distance threshold was set to 150m (identical to 2D ray tracing). Numerical results are shown in Table 3. An example of selected walls is illustrated in Figure 10d.

Method	Total # of visible walls	Avg # of cam. per wall	Computing time
10x10 rays	1349 (11.8%)	4.36	3min51s
20x20 rays	1604 (14%)	4.49	11min45s
10x10 rays αmax = 3*Π/8	1032 (9%)	4.55	2min49s
20x20 rays αmax = 3*Π/8	1213 (10.6%)	4.56	8min25s

Table 3. Results of 3D ray tracing

5.3 Discussion

As expected from a 3D-based approach, the walls located at the background can be textured if they are high enough. Fewer texture images are selected with the 3D ray-tracing approach than with the z-buffer approach, but they generally have a better quality. It is not surprising as ray tracing is not a dense approach and most small wall textures are naturally discarded. In the example of Figure 10e, only the relevant facade of the high building located at the back of the block was selected as visible. Figure 11 shows another example of distant facade that can be textured only with a 3D approach.
The additional pre-selection criterion on angles removes sidelong walls, which are usually seen by few cameras. It improves the relevance of the selection by discarding walls with a poor texture resolution. Using 20x20 rays instead of 10x10 rays significantly increases the total number of visible walls, but further tests are needed in order to find out whether these additional walls can be textured with a good enough quality. Importantly, as each wall is processed in turn, the texturing stage can be performed without requiring the complete processing of the path.
The computing time is intermediate between 2D ray tracing and 3D z-buffering. In our implementation many calculations are redundant. A spatial division of the space could be performed in order to make use of object-space coherence and accelerate ray tracing (Glassner, 1984; Jevans and Wyvill, 1989).

6. CONCLUSION AND PERSPECTIVES

The 2D approach is satisfactory in most cases, and it is fast, simple and easy to implement. However any building located behind another cannot be textured.
The 3D approaches provide more exhaustive wall textures, including texture images for high building walls located at the back of lower buildings. The use of the 3D dimension makes the visibility estimation closer to ground truth, and the selection process more efficient. Although both ray-tracing and z-buffer techniques can be implemented very efficiently, the approach based on 3D ray tracing is a good compromise to achieve a relevant selection. It also seems important to prefer a wall-by-wall analysis, as further texturing stages can then be performed without requiring the complete processing of the path. The z-buffering technique could be considered if the resulting depth image is a valuable source of information in further steps.
The main constraint for the 3D approaches is obviously the availability of a 3D building database. Given a 2D map, the 3D information can be derived from a correlation-based or LIDAR Digital Elevation Model, or even from the analysis of architectural plans or building permits. In our opinion, a coarse 3D city model is sufficient to significantly improve the relevance of the texture selection.

We are now working on refining the selection with texture quality criteria rather than just visibility. The texture quality

depends on its resolution and varies with the distance from the camera to the facade.
Another possible evolution is to use additional 3D information to predict occlusions. A Digital Terrain Model could be used to predict hidden parts due to hills or embankments (case of a hill masking buildings facades located on the other side of a square for instance). If available, a complete 3D city model including vegetation and detailed building roofs would help better estimate the visibility of a given façade. More generally, an environment mask as in described in (Wang et al., 2002) could be introduced.
Another parameter to take into account is the uncertainty on the GPS/IMU data which introduces an uncertainty on the camera position and direction. In order to guarantee a complete selection, a simple solution would be to dilate each wall polygon by the maximal distance induced by the positioning uncertainty. In a similar way, the influence of the input 3D model accuracy should be investigated.
For this particular study, only synthetic data have been used. In the future we will be working on real data, and the influence of both the positioning error and the 3D model accuracy will be studied. Figure 9 gives an idea of what we would like to automatically achieve at a large scale. Note that the side facade located at the top right of the image cannot be textured if the image selection process is only 2D-based.

Figure 9 – 3D virtual view of the historical centre of Rennes

REFERENCES

Allène, C., Pons, J.P. and Keriven R., 2008. Seamless image-based texture atlases using multi-band blending. *Pattern Recognition, ICPR 2008.*

Bentrah, 0., Paparoditis, N., Pierrot-Deseilligny, M., 2004. Stereopolis : An Image Based Urban Environments Modelling System. In *International Symposium on Mobile Mapping Technology (MMT)*, Kunming, China, March 2004.

Brun, X., Deschaud, J.E. and Goulette, F., 2007, On-the-way City Mobile Mapping Using Laser Range Scanner and Fisheye Camera, In *International Symposium on Mobile Mapping Technology (MMT)*, Padua, Italy2007.

Früh, C., and Zakhor, A., 2003. Constructing 3D City Models by Merging Aerial and Ground Views. IEEE Computer Graphics and Applications, 23(6), Nov. 2003.

Früh, C. and Zakhor, A., 2004. An Automated Method for Large-Scale, Ground-Based City Model Acquisition. *International Journal of Computer Vision*, 60(1), pp. 5-24.

Glassner, A., 1984. Space subdivision for fast ray tracing. *IEEECG&A*, 4(10):15-22, Oct. 1984.

Goulette, F., Nashashibi, F., Abuhadrous, I., Ammoun, S. and Laurgeau, C., 2007. An Integrated On-board Laser Range Sensing System for On-the-way City and Road Modelling. In *ISPRS Comm. I Symposium,* Marne-la-Vallée, France, 2004.

Greene, N., Kasse, M., Miller, G., 1993. Hierarchical Z-buffer visibility. In *Proc. Of the 20th conf. On Computer graphics and interactive techniques,* Anaheim, CA, 1993.

Haala, N., 2004. On the refinement of urban models by terrestrial data collection. In *XXth ISPRS Congress, Vol. 35, Part B,* Istanbul, Turkey, 2004.

Hunter, G, 2009. Streetmapper mobile mapping system and applications in urban environments. In *ASPRS Annual Conference*, Baltimore, USA, 2009

Jevans, D. and Wyvill, B. Adaptive voxel subdivision for ray tracing. *Proc. Graphics Interface '89* , 164-172, June 1989.

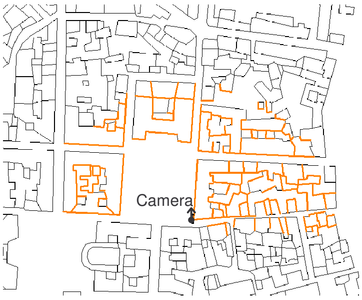
Mrstik, P., and Kusevic, K., 2009. Real Time 3D Fusion of Imagery and Mobile Lidar, *ASPRS Annual Conference*, Baltimore, USA, 2009.

Pénard, L., Paparoditis, N. and Pierrot-Deseilligny, M., 2005. 3D Building Facade Reconstruction under Mesh Form from Multiple Wide Angle Views, In *IAPRS vol. 36 (Part 5/W17),* 2005.

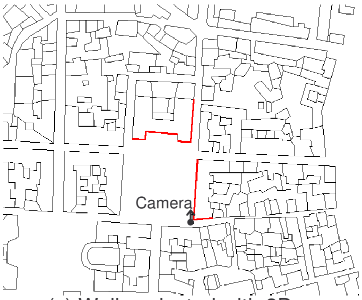
Strasser, W. *Schnelle kurven und Flächendarstellung auf graphishen Sichtgeräten*, Ph.D. Thesis D83, Technical University of Berlin, Germany, 1974

Wang X., Totaro S., Taillandier F., Hanson A. R. and Teller S., 2002. Recovering Facade Texture and Microstructure from Real-World Images. *Proc. of the 2nd International Workshop on Texture Analysis and Synthesis in conjunction with ECCV'02, pp. 145-149*, Copenhague, Denmark, June 2002.

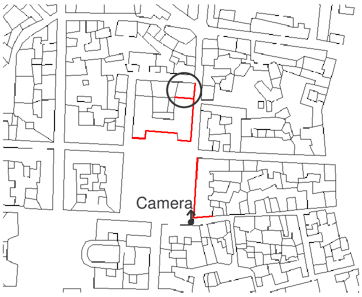
(a) Building heights
(blue=low, orange=high)



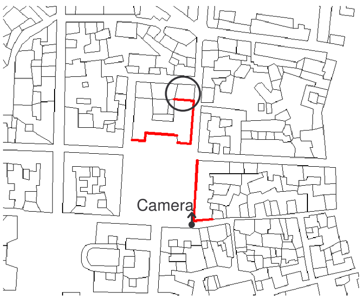
(b) Pre-selected building
walls (orange)



(c) Walls selected with 2D
ray tracing (red)



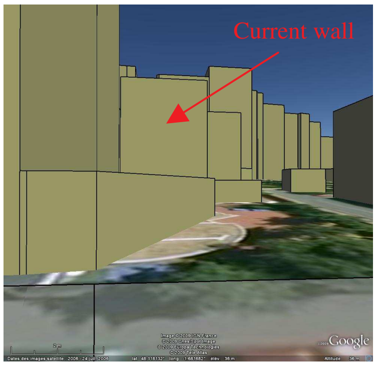
(d) Walls selected with 3D
z-buffers (red)



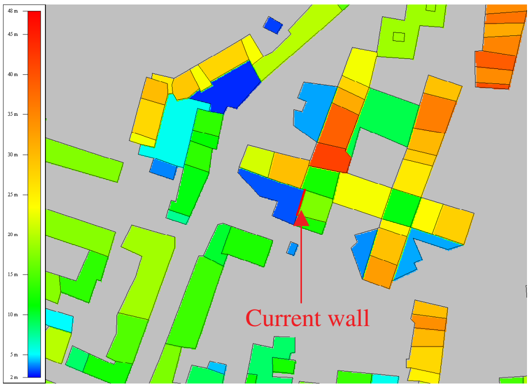
(e) Walls selected with 3D
ray tracing (red)

Figure 10. Results of the various selection methods

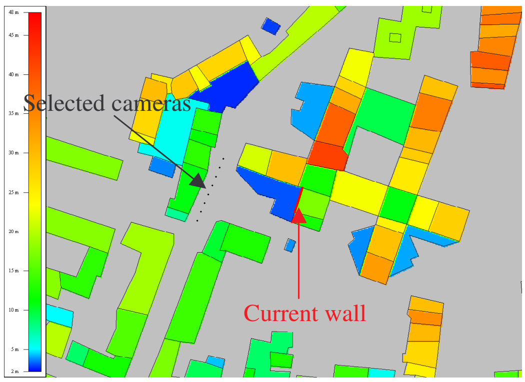
(a) Real view



(b) Virtual view of the input 3D model



(c) Result of the camera selection by 2D ray-tracing:
no camera was detected for the current wall



(d) Result of the camera selection by 3D ray-tracing:
8 cameras were found viewing the current wall.

Figure 11. Example of a low building masking a façade