

3D MODELLING OF A TOWN SCALE MODEL

C. Chevrier^a, K. Jacquot^a, J. P. Perrin^a

^a CRAI, National School of Architecture of Nancy, 2 rue Bastien Lepage, 54000 Nancy, France
(chevrier, jacquot, perrin)^a@crai.archi.fr

KEY WORDS: 3D modelling, parametric modelling, architecture, cultural heritage, photogrammetry, lasergrammetry.

ABSTRACT:

This paper presents a new “top-to-bottom” method for the 3D modelling of towns. This method was first tested in the case of a town scale model made of wood and paper. One of the applications that will be made with the 3D model is a web application. The model has to be light but reliable. From skeletons of the roofs extracted from pictures with PhotoModeler, we automatically build the 3D model of the town with the following points automatically handled: classification of the segments and of the kind of slopes, incoherences due to precision errors between segments of a roof and between roof outlines, overhang parameter values. Results are promising



Figure 1: Scale model of Toul

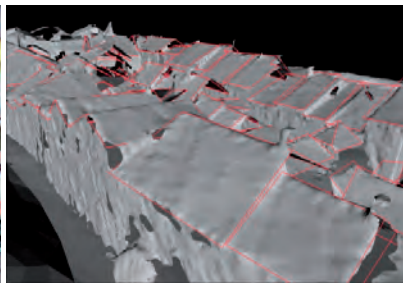


Figure 2: 3D segments on the laser data

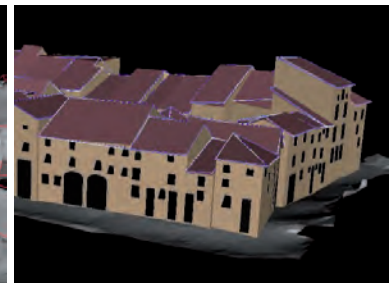


Figure 3: Automatic city block reconstruction

1. INTRODUCTION

The needs of 3D models are currently numerous for various uses: scientific studies (evolution of building deformities-vaults for instance, digital terrain model, petroleum reserves, volcano studies, etc), virtual visits (ancient or destroyed cities or monuments), historical studies of towns (scale models of town, hypothesis of building construction), illumination simulation projects of historical monuments, etc.

We work for the “Musée des plans-relief” (Museum of town scale models) in Paris and for the “service régional de l’inventaire général de Lorraine” (historical research centre) on the 3D modelling of the scale model of the town of Toul in France (Figure 1). The scale model of Toul was built in the 1840s and modified in the 1860s. It is composed of 20 pieces (called tables). The biggest table is the one containing the town (2.31 x 2.23m). The other tables contain the countryside. We were asked to model the town table and a table from the countryside (1.50 x 2.50).

Why model the scale models?

- Interactive terminals for the museum: there is no visibility of the town parts of the scale models because they are too big. They are behind glass to avoid dust deposit.
- Web application: navigation in the virtual model will be a means to access further information via the model.
- Most of the scale models are stocked in boxes where nobody can see them. It is the case for the scale model of Toul. Only 26 of them are exposed in the museum.

- For the researchers in history.
- For cultural heritage conservation.

Research problems are related to this project. They are at the mid-point of two research fields: photogrammetry and lasergrammetry on the one hand and 3D parametric modelling of architectural elements on the other hand. It is still unthinkable to obtain automatically a 3D scene composed of architectural parametric elements. However it is possible to automate some steps with a restricted user help and an adapted user interface.

In this paper, we present the method we have conceived and developed for the 3D modelling of scale models of towns. After a presentation of related works (part 2), part 3 explains the context of the project. Then the principles of the method are explained in part 4. Part 5 deals with the parameters and constraints we have identified for the creation of the architectural elements in the scale model. Some results are exposed in part 6. Finally, we conclude and present future work in part 7.

2. RELATED WORKS

Several works are undertaken in modelling scale models of cities. The model of Prague by Antonín Langweil in 1826-1837 is one of these projects. It involved a large team, more than a hundred people for just the 3D model reconstruction step. The digitizing was accomplished easily because the model can be divided into 52 parts (1.6 x 1m for the biggest with a scale of 1/480). It is in much better shape than the model of Toul but the buildings' textures contained thin structures of drawing in

Indian ink (the measured details corresponded to roughly 1200 DPI) making textures and their digitalisation more delicate than ours. The model was carried out thanks to the collaboration of Autodesk and the use of photogrammetric software adapted and developed especially for this project (Sedlacek 2009, [7]).

In the 3D modelling of Beijing city scale model (Zhu et al, 2009), the authors use stereo image pairs of the object to improve the quality of edges of the scanned model. The point clouds are improved but no modelling is made and no semantic is given to the model. In our project each building needs to be identified for the web application with links anchored to buildings.

In the 3D modelling of Rome scale model [8] (Guidi 2005), models of complex buildings were created with Maya [5] and the others were computed with the help of procedural and parametric modelling techniques (city Engine [11], Dylla 2009). Some research in photogrammetry and lasergrammetry create meshes from either laser point clouds or image-based point clouds and improve the quality of the edges and surfaces (Hoppe 2009, Barazzetti 2009, Fisher 2003, Zhu et al 2009, Sedlacek 2009). The main drawbacks of these methods remain the large amount of data (a simple surface like a cylinder will be composed of many points), the lack of semantic means (all objects are grouped in one big mesh), the lack of accuracy of edges (due to the sampling process of a laser scanner), absence of parts of the model (some parts are always hidden by foreground objects).

Other research extract geometrical data (roof planes) from point clouds (acquired from either laser scanning or images) and then build the 3D model of the houses from top to bottom (Vosselman 2001, Tse 2005, Tarsha-Kurdi, 2007, Boulaassal, 2009, YingYang 2010, Rottensteiner 2005, Maas 1999). (Pu and Vosselman, 2009) manage not only to rebuild automatically the 3D model of houses but also their openings.

As far as computer vision is concerned, extraction of segments or regions from images are also another research field of many teams (Werner 2002, Suveg 2002, Heuel 2000, Jaw 2008, Ameri 2000, Taillandier 2004...). (Hendrickx 1997, Koehl 2007, Moons 1998) use geometric properties to label each segment of a roof. Koehl divides data in tree layers: outline segments, every segment inside the outline and the roof slopes. He uses TIN (Triangulated Irregular Network) algorithms to triangulate roofs but problems involving complex roofs still appeared.

Few top to bottom methods deals with wall interpenetrations of juxtaposed buildings with constraints (Suveg 2002, Lafarge 2006), relationships between neighbouring roofs (gaps or overlapping slopes) and overhangs or verges.

On the other hand, there is research in architecture for the 3D modelling. Two kinds of research can be specified: 3D models for conception goals and 3D models for patrimonial or archaeological goals (Chevrier, 2009). The use of architectural 3D parametric components allows a quick and accurate modelling. For example, once the roof is modelled, roof superstructures constitute an accuracy asset for photorealism issues. A parametric model (Brédif, 2007) specific to several superstructure types is a fast way to attain this aim. However the adjustment of the parameters of the virtual 3D model in order to fit the real one is tedious. Some components are easily identifiable, the presence or not of a keystone, a type of roof, etc. but the value of a geometric parameter (length, radius, etc.) cannot be close to reality without additional data like surveys, photogrammetric or scanned models.

In this paper we use a top-to-bottom method and we handle automatically the following points: classification of the segments and of the kind of slopes, incoherences (correction of parallelism, and interpenetration) due to precision errors between segments of a roof and between roof outlines, overhang parameter values. Then a Graphical User Interface allows

manual corrections and the adding other components to the 3D scene (openings, chimneys, belt courses, quoins, vegetation...).

3. CONTEXT OF THE PROJECT

The aim of the project is to get a virtual 3D model of the scale model of the town of Toul in France (Fig 4 and 5). Compared to a 3D modelling of a real size town some specificities can be noticed:

- The size of the houses is very small: the scale is 1:600 that means approximately buildings of 2 to 3 cm high, streets are about 1 to 2 cm wide with some alleys of less than 5 mm wide.
- There is just the essential on the scale model: no perturbing objects in the streets (trees, urban furniture).
- The scale model is in a bad state: papers are unstuck or missing, a lot of dust is on the model, slopes are skew surfaces (Figure 1)
- Digitalization has to be carried out without contact, purely on optical principles.
- Given the fact that some areas of the model are not easily accessible, many houses cannot be well documented.
- Moving the model without a skilled team to access some areas is impossible because of the size, the weight and the fragility of the model.

The following parts explain the problems we encountered with the acquisition of the data: laser data (part 3.1), pictures (part 3.2) and with the use of 2D documents (part 3.3).



Figure 4: Photograph of the scale model of Toul



Figure 5: Photograph of the town table of the scale model of Toul (2.31 m x 2.23m)

3.1 Acquisition of the laser data

We were not allowed to touch the model. After much discussion we were authorized to lay several sewing threads in surrounding streets. We encounter difficulties in accessing the inner blocks and thin roads (Figure 6) The laser data will only be used for the numerical terrain model (DTM) on which we will build houses and plant vegetation.

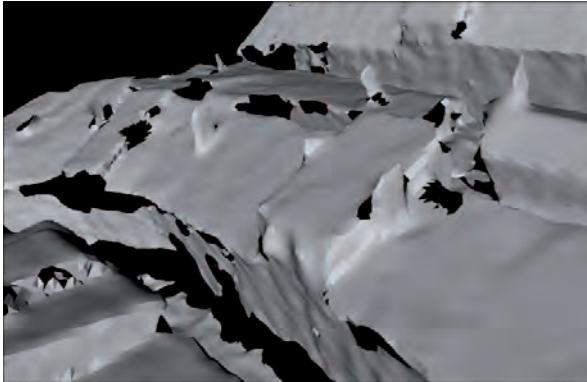


Figure 6: scanned parts realized with a precision of 0.5 mm. A resolution of 0.5 mm is a good compromise between time spent and quality.

3.2 Acquisition of the photographs

As mentioned, the size of the town part raised difficulties to access several city blocks. Due to the width of some streets or the density of buildings – especially in the centre of the model – photos did not show every part. Bad lighting conditions and bad contrast due to dust deposit increased the amount of time spent in taking pictures. Tests with PhotoModeler Scanner were not workable because of the bad quality of the photographs and of the bad state of the scale model itself.

3.3 No-Use of the 2D documents

Regarding the ground plans and elevation plans, there were some contradictions with the model because of lack of precision or modifications in the city during time lapse between the making of plans and the creation of the scale model. Actually, they were not ground plans but rather a hybrid of roof plans and ground plans. That's why many incoherences appeared making the use of these documents tricky.

While we wanted to create our model thanks to these documents, it appeared impossible to rely on them. Without ground plans and elevation plans, modelling relied only on photographs with the use of photogrammetry techniques. We have developed a “top to bottom method” specially adapted to scale models that can have irregular geometry and complex roof configurations.

4. PRINCIPLES OF THE METHOD

To create a 3D model, many tools are at our disposal. From Image-based process (ImageModeler [4] or PhotoModeler [6] software) to range-based modelling (3Dipsos [10], Geomagic [3] or polyworks [12]), even traditional geometry-based software (autoCad [2], archiCad [1]). Combining several technologies where they are best suited is one backbone of this work. Photogrammetry alone is not appropriate to build irregular shapes like sculptures or, in our case, ground relief

whereas laser scanning is an awkward choice to digitalize flat surfaces like roofs.

Thus, we have begun by recovering roof edges (part 5) thanks to PhotoModeler [6], a photogrammetric software. At the same time, the scanned ground geometry (digital terrain elevation DTE) has been manually treated with Geomagic [3] to create the digital terrain model (DTM): all the parts above the ground (buildings and vegetation) have been removed and holes have been filled to get the entire terrain model.

Automatic treatments on the 3D segments of the roofs allow removing excess points or segments, correcting parallelism between edges and fusion between close points (part 6). Then an automatic 3D modelling of the roofs and walls can be performed taking into account some parameters and constraints like the planarity of the roof slopes (part 7). The DTM is used to set up the walls of the buildings (to find the ground polygon corresponding to the building). As the 3D model will be primarily used for a web navigation, we have chosen to replace the DTM corresponding to the roads and below the buildings with a Delaunay triangulation of the encompassing area with holes corresponding to the buildings. For this we use the free software developed by (Triangle [9]). Finally manual refinement can be applied locally to the 3D model to adjust or correct parameters and constraints (part 8).

In all steps of the process, it is possible to use specific user defined thresholds. Every step can be cancelled to try other values of the thresholds.

5. SEGMENTS CREATED WITH PHOTODELER

The first step is a photogrammetric method involving the manual digitising of a minimum number of data necessary for the treatments to follow. To obtain the finished house model only the roof skeleton is needed.

In the results from PhotoModeler, every surface is a skew surface, no line is perfectly parallel to another, roofs and walls of the houses can penetrate each other, etc. It is true that bad orientations in PhotoModeler or inaccurate point positions contribute to this kind of geometry but most of the model is so damaged (unstuck papers) that its geometry has become irregular. Nevertheless, we are able to use the accuracy of each PhotoModeler point in our further treatments to balance or advantage the position of some points or segments compared to others. Combined with automatic treatments (part 6), parametric objects and an adapted user interface, we can make the 3D model more accurate. The great majority of the roof slopes were designed planar surfaces because their realisation is easy. Skew surfaces are unusual but we keep them skew if they were designed skew or if their bad states do not allow us to recover their initial shape.

We work city-block by city-block with oblique pictures. The accuracy of the photometric data is around seven pixels for all the city-blocks. This high level is due to imprecise points we had to position although they were not seen in any picture or to points in blurred parts of the pictures. This accuracy corresponds to approximately half a millimetre. The segments are positioned automatically on the DTE with the help of three selected points in the DTM and the three corresponding points in the PhotoModeler data. This allows us to scale and orientate the segments. Manual refinement is possible if necessary.

6. AUTOMATIC TREATMENTS

6.1 Separating the total set of segments into small sets corresponding to a roof

The segment set is divided into small segments corresponding to a roof or sometimes a complex roof or a group of roofs. For a given segment, we look for all segments that are connected together: it means that they share a same point. If two roofs share a same point, there will be only one set of segments for the two roofs. This is not a problem as the following algorithms handle this case. We often have complex roofs and even a user does not really know how to separate pieces in several roofs (See Figure 7 for an example).

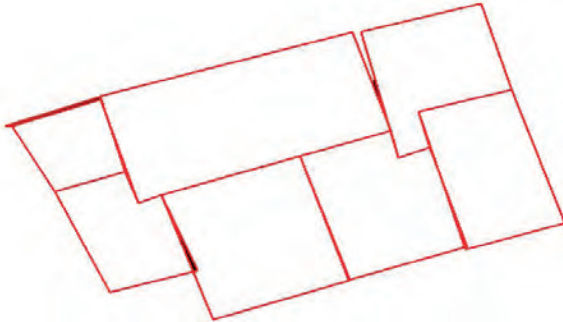


Figure 7: Complex roof structure detected as a unique roof. However the further classification and treatment correctly handle these cases.

In PhotoModeler, as we have seen in the previous section, there are sometimes exceeding points or segments. We then look for closed points (distance less than a given value). We only keep the most accurate point and also remove the double segments.

6.2 Classification of the roof segments

This step aims to identify the outline and the slopes of the roof. It also determines the type of each segment of a roof: ridge, roof edge, angle or valley, roof break, head edge and eave (Figure 8). Each slope is also classified for automatic corrections and material assignment: a normal slope (slope, hip, upper slope, break, half pipped) or a break.

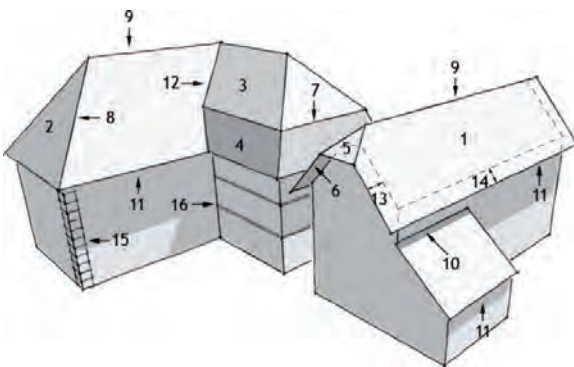


Figure 8: Various kinds of segments for the classification. (1:slope; 2: hip; 3: upper slope; 4: break; 5: half pipped; 6: ridge edge; 7: roof break; 8: angle; 9: ridge; 10: head edge; 11: eave; 12: valley; 13: verge; 14: eave overhang; 15: quoin; 16: belt course)

To determine the outline of the roof (encompassing polygon), we use an algorithm based on the detection of a bounding polygon of a set of points:

Algorithm

Outline O

Point B = the point with the lowest horizontal coordinates (x and z). Y axis is the vertical line.

Segment S = X axis

While there are segments in the set and we have not returned to point B **do**

 Among the set of segments, select the one (called S2) that has the smallest angle with the segment S

 Remove S2 from the segment set

 S = S2

 Add S2 to O

endWhile

To avoid bad results with complex roof shapes due to bad alignments of segments (overlapping slopes in Figure 7), we had to make the algorithm robust by taking into account these cases in the computation of the angle.

Once the outline of the roof has been computed, angles and valleys are easily separated because there are not horizontal segments (with a given threshold). The remaining segments are ridges and roof breaks. In the same way, we identify roof edges in the outline of the roof. The remaining segments are eaves. For roofs with no other segments than the segments of the outline (it means roofs composed of only one slope), we identify a head edge among the edges as the higher segment (or highest segments in some cases).

Almost the same algorithm is used to identify all the segments that belong to each roof slope. The algorithm is adapted based on the consideration that a segment belonging to the outline can only be used once whereas the other segments can be used twice (they separate two roof slopes).

Algorithm

While there are segments in TSS **do**

 Create a slope

 Segment S = first segment in the total segment set of the roof (TSS)

If S is a segment from the outline **then** remove S from TSS

Elseif the number of times we have used S2 is one **then**

 Remove S2 from the segment set

Else Mark the S as used once

Endif

 Add S to the slope

While we have not returned to the beginning point **do**

 Among the set of segments (TSS), select the one (called S2) that has the biggest angle with the segment S

if S2 is a segment from the outline **then**

 remove S2 from TSS

Elseif the number of times we have used S2 is one **then**

 Remove S2 from the segment set

Else

 Mark S2 as used once

Endif

 S = S2

 Add S to the slope

endWhile

endWhile

The roof slopes are then automatically classified as normal or break slope. This allows assigning a specific material to the different parts of the roof during the modelling process. Indeed a break slope is usually in slate whereas a normal slope is usually made of roof tiles. In case of errors in the classification, manual correcting can be applied: The user interface lets us access each segment or each slope in order to modify the kind.

6.3 Corrections of parallelism and fusion of points

In order to obtain realistic shapes similar to the physical model, we have to constrain roof geometries inherited from bad orientations in PhotoModeler. The other goal of this measure is to prevent building imbrications.

The first kind of corrections we make are parallelism corrections. This is done in 2D (XZ plane ; Y axis is vertical) First in a same roof outline, neighbouring roof-edge segments that are almost parallel (to a given threshold angle) are put perfectly aligned. Then each segment of the outline is compared to the neighbouring outline segments of other roofs.

If the two segments have an influence on each other we superimpose them or we make them parallel according to a certain threshold: if the distance between the two segment midpoints is less than a certain value (0.05 cm for example in our case) then we superimpose them. If the distance between the two segment midpoints is less than a certain value (0,4 cm for example) then we make them parallel.

To determine if two segments have an influence on each other, we proceed as follows: first we consider that one roof can have an influence on another roof if the distance of their weighted centres is less than a given threshold. This eliminates quickly distant roofs. Then we apply the following algorithm:

Algorithm

For each segment S1 of an outline roof R1 **do**

For each segment S2 of an outline roof R2 **do**

If the angle between the two segments is less than a given threshold and the distance between their midpoints is less than another threshold **then**

 Project the points of S2 on the line defined by S1

 This defines the length of the common part on the line

 Float k1 = length of the common part / length of S1

 Float k2 = length of the common part / length of S2

 Float k = k1 + k2

If k > another threshold **then**

 S1 and S2 influence each-other.

Endif

Endif

Endfor

Endfor

To determine if S1 has to be aligned or superimposed on S2 or the opposite or if both segments have to be moved along a mid median, we use point precision and some criteria based on the length of each segments: a small segment (corresponding to a sloping roof for example) will be moved to be aligned to the other segment (the wall of a house for example).

If two segments have an influence on each other and if these segments are roof edges we also correct other connected roof edge segments.

This constraint of parallelism also allows us to determine the overhang values. At the beginning of the process every roof-edge segment has a user-defined value and every eave segment has another user-defined value. When two segments are superimposed, both the overhang values are set to 0 (Figure 9).

When two segments are put parallel, if the roof R1 and roof R2 have a common part, we study their relative position in 3D (not only of the 2 segments but also of the connected roof-edge segments if necessary): if the polylines thus defined intersect each other, both the overhang values are kept (parts of the segments of the two buildings have overhangs and parts do not have any overhangs). In the other case we determine the polyline that is above the other: the segments of the above polyline keep an overhang value that is equal to the distance between the two segments and the segments of the other polyline have an overhang value set to 0 (Figure 10).

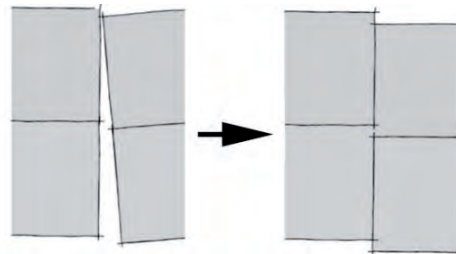


Figure 9: Constraints of parallelism to determine overhangs

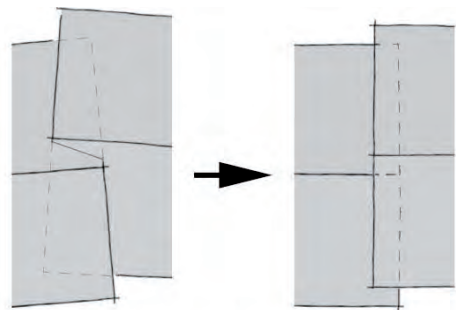


Figure 10: One polyline is above the other: only the overhang of the highest polyline is different to zero.

Thresholds are chosen to maximize the right corrections without wrong corrections. After the process, if the result doesn't suit the user, he can cancel the corrections and apply again the process with other thresholds.

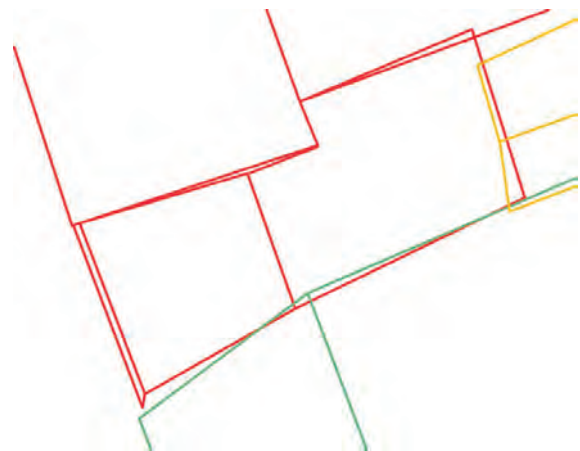


Figure 11: Interaction of three different roof skeletons in three different colours. One can see the parallelism errors between segments.

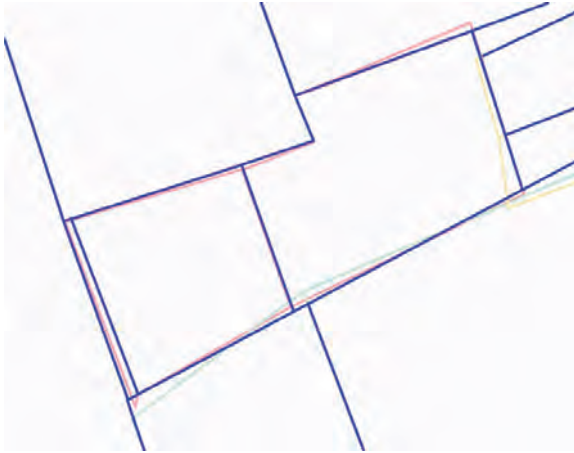


Figure 12: Automatic corrections of parallelism between segments of a same outline and between outlines in the cases of the three outlines of Figure 11.

7. AUTOMATIC 3D RECONSTRUCTION OF HOUSES AND DIVIDING WALLS

Based on the classification of the segments composing a roof, we can now automatically create the 3D model of the corresponding house. Parameters and constraints are defined for all buildings for this step but each parameter or constraint will be able to be modified for each house in the next step if necessary (manual refinement). The main parameters and constraints taken into account in this step are: a value for the thickness of the roof, planarity of the roof slopes, horizontal ridges. Parameters have default values that are the most common values encountered. Constraints are the most common. The constraints are first taken into account to modify the position of the points. Each ridge is set to horizontal (taking into account the accuracy of the points or according to the mid point of the segment).

Then, for each roof slope, the planarity is applied: the ridge is supposed to be well positioned, the most precise point among the other points of the slope is selected to define the supporting plane of the slope. The other points are then vertically projected onto this plane to be positioned.

Algorithm:

```

ForEach segment of the roof outline do
  - Compute the offset points due to the roof thickness,
    perpendicular to each roof slope it belongs to. The position of a point
    depends on the offsets due to the two outline segments that share that
    point.
  - Compute the offset points due to the overhang of each segment.

```

```

ForEach roof slope do
  Create the polygon of the slope
  ForEach segments of the slope do
    If the segment belongs to the roof outline then
      Create the polygon corresponding to the roof thickness
      If the overhang of the segment is > 0 then
        Create the polygon corresponding to the overhang
      Endif
    Endif
  Endfor
Endfor

```

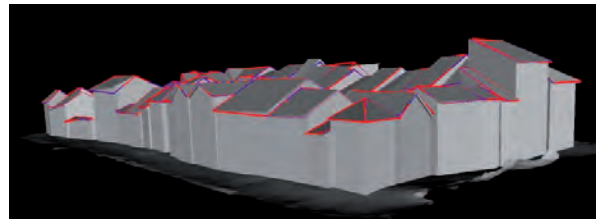


Figure 13: Automatic 3D model obtained from the roof skeletons. Slopes are initially set plane and ridges are set horizontally but the user can modify these parameters to adjust the model.

8. USER INTERFACE FOR THE ADJUSTMENT OF PARAMETERS AND CREATION OF OTHER OBJECTS

As in any automatic process, most of the corrections are pertinent, but in some cases they are not pertinent or some problems have not been corrected whereas they should have been. This is due to the values given to the thresholds used during the automatic process. We have then developed a user interface to apply locally some constraints or modify values of the parameters. It is thus possible to select a segment and to apply locally some treatment (the same as the automatic ones but only for this segment with user-chosen thresholds). For wrong automatic corrections, it is possible for a specific segment to cancel the automatic treatment. Values of the parameters can be adjusted for each segment or each roof. Finally it is also possible to modify the position of a point if the constraints failed for that specific point.

8.1 Parameters and constraints

Parameters for segments are: type of segment (ridge, edge, angle or valley, break, head edge and eave) and an overhang value. Parameters for a roof are: roof thickness, roof slope planarity, horizontal roofing, eave overhang, parallel roof edges and ridges.

8.2 Adding other objects to the model

Openings, dormer windows, belt courses, chimneys are all treated thanks to parametric objects. We distinguish objects attached to the walls (openings, belt courses) and objects attached to the roof (chimneys, dormer windows).

We have identified all the kinds of objects and realized a 3D model of each kind with a modeller. Each model is bounded in a square 1 by 1 cm and 0.1 cm deep. To position an object on a wall we then specify the type of object, the wall of the building on which the object must be stuck, the height, width and depth of the object (3 scale factors applied to the model), the x position along the wall, the y position in the vertical direction. We can also specify other parameters to simplify duplication of a same object:

- number of objects along the wall, horizontal interval between objects, boolean that indicates if the objects are centred along the wall with a possible lateral interval.
- number of objects vertically, vertical interval between objects,
- index of a removed object from the duplication: another kind of object takes the place of the previous object (generally a door instead of an opening)
- boolean that indicates if the object is as high as the wall and a boolean that indicates if the object is as wide as the wall (generally used for belt courses).

To position an object on a roof, once the kind of object has been chosen, and its position calculated, the bottom points of the objects are projected onto the roof slope to produce a correct layout. For complex objects (like a construction on the roof of another one), a parametric object is positioned on the roof.

9. RESULTS

The town was decomposed in blocks. Each block was treated separately in PhotoModeler and then in our software for automatic treatment, manual adjustments and for the adding of other components. The town is approximately composed of 4000 buildings. We classified the buildings of 20 blocks according their roof shape (see Figure 14).

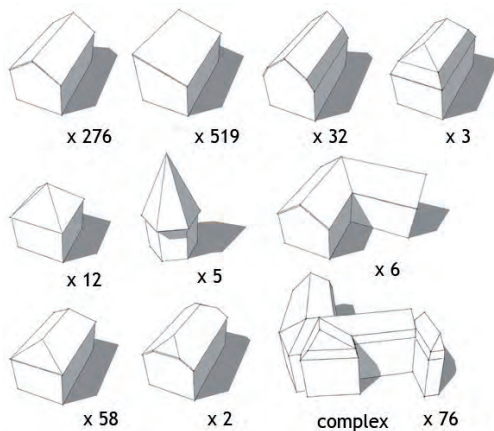


Figure 14: Classification of the roofs found in the scale model for 20 blocks and 898 buildings

Figure 15 and Figure 16 show examples of blocks with openings and chimneys.

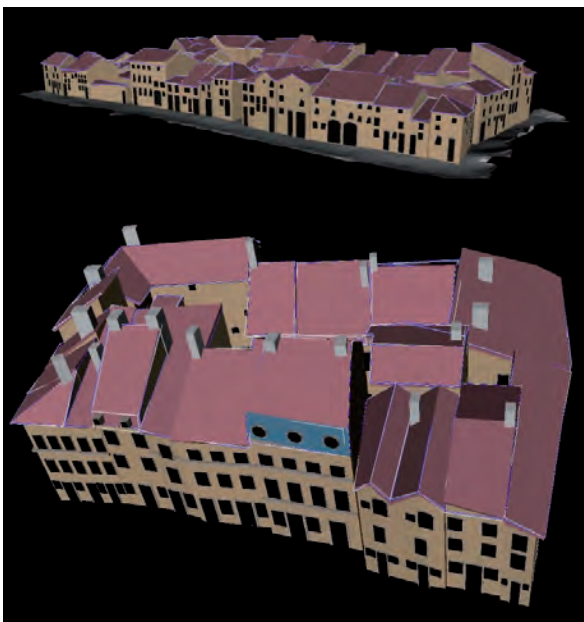


Figure 15: 3D model of blocks with elements added to the automatic creation: openings, belt courses, chimneys...



Figure 16: A view of the 3D model of several blocks

10. CONCLUSION AND FUTURE WORK

In this paper we have exposed the methods and algorithms we have used for the reconstruction of a scale model of a town. This scale model was made in wood and paper. Its bad state obliged us to have the possibility of keeping skew surfaces. An automatic and accurate classification of the segments of the roof skeletons allowed us an entirely automatic 3D modelling with automatic determination of the values for the mains parameters. The user only interfered when errors occurred or adjustments had to be made.

In the future, several important points could be improved or tested:

- Try our methods on existing towns (aerial photographs and laser data).
- Improve the step of acquiring the segments of each roof. This step is carried out manually with PhotoModeler software. For this we plan to collaborate with researchers in computer vision and photogrammetry.
- Implement automatic techniques for the positioning of textures.
- make use of the DTE to adjust the roof geometry with an automatic detection of planes as in (Tarsha-Kurdi, 2007) for example.

REFERENCES

References from Journals:

Boulaassal, H., Landes, T., Grussenmeyer, P., 2009. Automatic extraction of planar clusters and their contours on building façades recorded by terrestrial laser scanner. *International Journal of Architectural Computing (IJAC)*, Volume 7, Number 1, January 2009, pp. 1–20.

References from Other Literature:

Ameri, B. and Fritsch, D., 2000. Automatic 3D building reconstruction using plane-roof structures. *ASPRA Conference*, Washington, DC.

Barazzetti, L. Remondino, F., Scaioni, M., 2009. Combined use of photogrammetric and computer vision techniques for fully automated and accurate 3D modeling of terrestrial objects.

- Videometrics, Range Imaging and Applications X, Proc. of SPIE Optics+Photonics, Vol. 7447, 2–3 August, San Diego, CA, USA
- Brédif, M., Boldo, D., Pierrot-Deseilligny, M. and Maître, H., 2007. 3D Building Reconstruction with parametric roof superstructures. In: IEEE International Conference on Image Processing, San Antonio, USA, September 2007.
- Chevrier, C., Perrin, J.P., 2009: Generation of architectural parametric components, CAAD Future conference, June 17–19, Montreal, Canada, pp. 105–118.
- Dylla K., Müller P, Ulmer A., Haegler S. and Frischer B.. 2009. Rome Reborn 2.0: A Framework for Virtual City Reconstruction Using Procedural Modeling Techniques. Proceedings of Computer Applications and Quantitative Methods in Archaeology (CAA).
- Fisher, R.B., Solving architectural modelling problems using knowledge, 3-D Digital Imaging and Modeling, 2003. Proceedings. Fourth International Conference (343–351).
- Guidi, G., B. Frischer, De Simone, M., Cioci, A., Spinetti, A., Carasso, I., Loredana, I., Russo, M. and Grasso, T., 2005. Virtualizing Ancient Rome: 3D Acquisition and Modeling of a Large Plaster-of-Paris Model of Imperial Rome, Videometrics VIII, edited by J.-Angelo Beraldin, Sabry F. El-Hakim, Armin Gruen, James S. Walton, 18-20 January 2005, San Jose, California, USA, SPIE, vol. 5665, pp. 119–133.
- Hendrickx M., Vandekerckhove J., Frere D., Moons T. and Gool L. V., 1997. 3D reconstruction of house roofs from multiple aerial images of urban areas. In: IAPRS, Vol. 32 Part 3-4W2, Stuttgart, September 17–19 1997, pp. 88–95.
- Heuel, S., Frstner, W., and Lang, F., 2000. Topological and geometrical reasoning in 3D grouping for reconstructing polyhedral surfaces. In ISPRS, volume XXIII, pages 397.404, Amsterdam.
- Hoppe C. and Krömker S., 2009: Adaptive meshing and detail-reduction of 3D-point clouds from laser scans. , proceedings of ISPRS Workshop 3D ARCH 2009, Trento (Italy), February 25-28.
- Jaw, J.J. and Cheng, C.C., Building roof reconstruction by fusing laser range data and aerial images. In Proc. ISPRS Congress, pages 707–712, Beijing, China,.
- Koehl, M. and Darwish, O. , 2007: Construction et intégration de maquettes 3D dans un SIG, Conférences SIG 2007, Versailles, France.
- Lafarge, F., Descombes, X., Zerubia, J., and Pierrot-Deseilligny, M., 2006. Automatic 3D building reconstruction from DEMs: an application to PLEIADES simulations. In IAPRS, Volume XXXVI Part A, pp. 129–136.
- Maas, H.G. and Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data, ISPRS Journal of Photogrammetry and Remote Sensing 54 (2-3): 153–163.
- Moons, T., Frère, D., Vandekerckhove, J. and Van Gool, L.J., Automatic Modelling and 3-D Reconstruction of Urban House Roofs from High Resolution Aerial Imagery, Proc. 5th European Conference on Computer Vision, Freiburg, Germany, pp. 410–42
- Rottensteiner, F., Trinder, J., Clode, S. and Kubik, K., 2005. Automated delineation of roof planes from LiDAR data, ISPRS Workshop Laser scanning 2005, Enschede, the Netherlands.
- Sedlacek, D. and Zara, J. 2009. Graph Cut Based Point-Cloud Segmentation for Polygonal Reconstruction. In Proceedings of the 5th international Symposium on Advances in Visual Computing: Part II, Las Vegas, Nevada.
- Suveg, I. And M.G. Vosselman, 2002, Automatic 3D building reconstruction, Photonics West 2002: Electronic Imaging, Volumes 4657 - 4677, SPIE, San Jose, California, 19 – 25 January 2002 pp. 59 – 69.
- Taillandier, F., Deriche, R., 2004. Automatic buildings reconstruction from aerial images: a generic Bayesian framework. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 35 (Part B3).
- Tarsha-Kurdi, F., Landes, T. and Grussenmeyer, P., 2007. Hough-transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data, ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo, Finland.
- Tse R.O.C, Dakowicz M., Gold C. M., and Kidner D.. Building reconstruction using LIDAR data. In Proceedings 4th ISPRS Workshop on Dynamic and Multi-dimensional GIS, pages 156–161, Pontypridd, Wales, UK, 2005.
- Vosselman, G. and Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol 34, part 3/W4, October 22–24, Annapolis, MA, USA, pp. 37– 44.
- Werner, T. and Zisserman, A., New Techniques for Automated Architectural Reconstruction from Photographs, ECCV02(II: 541 ff.).
- Ying Yang M. and Förstner W. (2010): Plane Detection in Point Cloud Data. Technischer Report, Institute of Geodesy and Geoinformation, Department of Photogrammetry, 2010.
- Zhu, L., Ma, G., Mu, Y. and Shi, R., 2009. Reconstruction 3d-models of old Beijing city structured light scanning, 22nd CIPA Symposium, October 11–15, Kyoto, Japan.

References from websites:

- [1] archiCad: www.aricad-cao.com, (accessed august 2010)
- [2] AutoCad: www.autodesk.fr (accessed august 2010)
- [3] Geomagic: <http://www.geomagic.com> (accessed august 2010)
- [4] ImageModeler: <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=11390028> (accessed august 2010).
- [5] Maya: <http://usa.autodesk.com> (accessed august 2010).
- [6] PhotoModeler: <http://www.photomodeler.com> (accessed august 2010)
- [7] Scale Model of Pragues: http://www.langweil.cz/index_en.php (accessed august 2010)

[8] Scale model of Rome: www.romereborn.virginia.edu
(accessed august 2010)

[9] Triangle: <http://www.cs.cmu.edu/~quake/triangle.html>
(accessed august 2010)

[10] Trimble:
<http://www.trimble.com>
(accessed august 2010).

[11] City Engine: <http://www.procedural.com> (accessed august 2010).

[12] Polyworks: <http://www.innovmetric.com> (accessed august 2010)