2005-10

SAMATS: Edge Highlighting and Intersection Rating Explained

Joe Hegarty
Dublin Institute of Technology, joe@dmc.dit.ie

James Carswell
Dublin Institute of Technology, jcarswell@dit.ie

2005-10-01

SAMATS:edge highlighting and intersection rating explained

Joe Hegarty
Dublin Institute of Technology, joe@dmc.dit.ie

James D. Carswell
Dublin Institute of Technology, jcarswell@dit.ie

SAMATS – Edge Highlighting and Intersection Rating Explained

Joe Hegarty and James D. Carswell

Digital Media Centre, Dublin Institute of Technology, Aungier St., Dublin, Ireland.
{joe@dmc.dit.ie, jcarswell@dit.ie}

Abstract. The creation of detailed 3D buildings models, and to a greater extent the creation of entire city models, has become an area of considerable research over the last couple of decades. The accurate modeling of buildings has LBS (Location Based Services) applications in entertainment, planning, tourism and e-commerce to name just a few. Many modeling systems created to date require manual correspondences to be made across the image set in order to determine the models 3D structure. This paper describes SAMATS, a Semi-Automated Modeling And Texturing System, which has the capability of producing geometrically accurate and photorealistic building models without the need for manual correspondences by using a set of geo-referenced terrestrial images. This paper gives an overview of SAMATS' components, while describing the Edge Highlighting component and the Intersection Rating step from the Edge Recovery component in detail.

1 Introduction

This research investigates building reconstruction technology for creating geometrically accurate, photorealistic 3D models from terrestrial digital photography for use in LBS (Location Based Services) applications. It is envisioned that the resulting 3D model output from this work be web-enabled and made available to subsequent LBS research endeavors (e.g. for archaeologists, town planners, tourism, e-Government, etc.). Being able to produce 3D building models using terrestrial imagery allows all users to exploit the future commercialization potential of web-based LBS, as demonstrated in [1].

[10] was the first to investigate the principle of structure from motion. [9] builds on these ideas using lines instead of points, although both require correspondences to be made manually across the image set. In fact the majority of semi-automated reconstruction systems require the user to make manual correspondences across the image set in order to reconstruct a model, which is generally a very time consuming task. [3] is one of the most robust systems using this approach which allows the user to create models using a set of block primitives and by setting constraints on those primitives. A more automated modeling approach involves the modeling of roofs using aerial imagery. Models produced in this way can produce structurally accurate models but fail to capture building façades accurately, although [5], [6], and [7] have looked into

the merging of façade textures with models produced from aerial imagery. [2] constructs a large set of 3D building models by using spherical mosaics produced from accurately calibrated ground view cameras fitted with GPS. Although highly automated, this system was limited to modeling simple shaped buildings by simply identifying the rooflines and extruding walls downwards.

SAMATS uses a novel approach to creating building models without the need for manual correspondences to be made. [11] is an example of extracting building and window edges without the need for manual correspondence, although a rough model of the structure being modeled is required in order for this system to work. No prior building model is required by SAMATS. The ability of SAMATS to remove the manual correspondence step found in most modeling approaches is achieved by having all images geo-referenced in the same reference frame. However, the acquisition of geo-referenced terrestrial images is still a serious bottleneck that does not have a straightforward solution. Currently public GPS will give an absolute accuracy of between 1 to 10 meters using a single receiver. This resolution is not technology bound but information restriction bound, with military GPS offering centimeter accuracy. As private industries or other governments create their own satellite networks these restrictions may no longer apply - making the acquisition of accurate geo-referenced imagery as simple as regular imagery. SAMATS does not solve the difficulties in acquiring geo-referenced imagery - it only investigates the usefulness of such imagery in the overall modeling process.

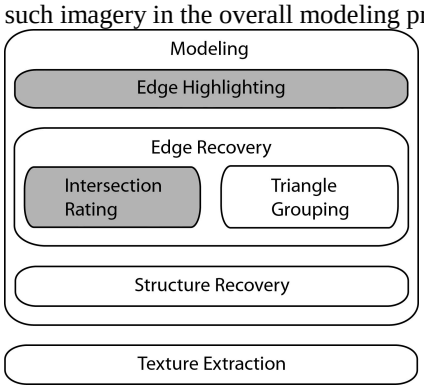

Fig. 1. SAMATS system diagram. The highlighted steps are the focus of this paper

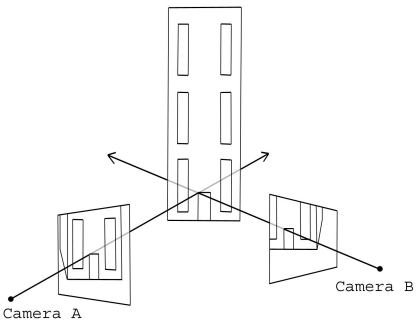

Fig. 2. Two point projections used to determine a point in 3-space

This paper gives an overview of the entire SAMATS system, while focusing on the Edge Highlighting component and the Intersection Rating step of the Edge Recovery component. For a detailed description of the other components refer to [4]. Figure 1 shows a systems overview of SAMATS.

2 Modeling

This section describes the process used to model the geometry of a building from a set of geo-referenced images using only simple edge highlighting by the user. The basic concept behind the modeling process is as follows; if one has two images of a scene taken from different locations, and the exact position and orientation of the camera is known for each image (i.e. the exterior orientation parameters $\mathbf{X_o}$ $\mathbf{Y_o}$ $\mathbf{Z_o}$ $\mathbf{\Omega}$ $\mathbf{\Phi}$ and $\mathbf{K}$) then the exact location of any point visible in both images can be determined. This is illustrated in figure 2.

The modeling process outlined in this section extends this idea by using triangle intersections to find edges rather than line intersections to find points. The modeling process can be split into three main steps; Edge Highlighting, Edge Recovery and Structure Recovery.

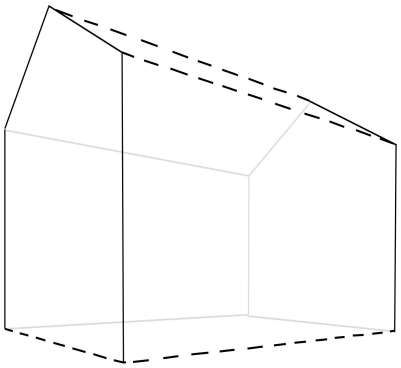

Fig. 3. House outline, primary lines solid black, secondary lines dashed black

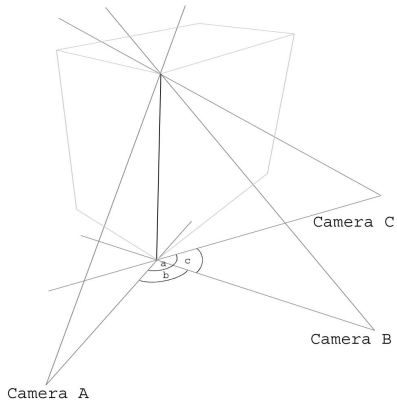

Fig. 4. For vertical edges, large disparity angles can be achieved

2.1 Edge Highlighting

Edge highlighting is the only manual step performed by the user in the modeling process. Primary lines and secondary lines are used to highlight edges in the images. Primary lines are used to recover the position of edges directly, determining the core structure of the model. They are responsible for the creation of every vertex in the final model. The endpoints of a primary line can be connected (having one or more primary or secondary lines sharing that endpoint) or unconnected (having no other lines sharing that endpoint). A secondary line is used to connect primary lines together and must have each of its endpoints connected to at least one primary line. In figure 3 the solid black lines represent primary lines while the black dashed lines represent secondary lines.

The reason the entire model is not defined by primary lines is because it is difficult to recover some edges given the input data. Primary lines are well suited to recovering the position of vertical edges because it is possible to create arbitrarily large angles of intersection about the vertical edge axis, as shown in figure 4. However, for horizontal edges near camera level it is not possible to create arbitrarily large intersection angles, making it difficult to recover the horizontal edges accurately since slight inaccuracies in the camera's intrinsic or extrinsic properties results in large errors in estimated edge location, see figure 5.

Secondary lines work by connecting primary lines, where the use of a primary line would be prohibitive, e.g. the horizontal base line of the building in figure 5. Since the primary lines will recover the vertical edges of the building, the secondary lines simply indicate to the system that these edges should be connected without trying the same recovery technique used for the primary edges.

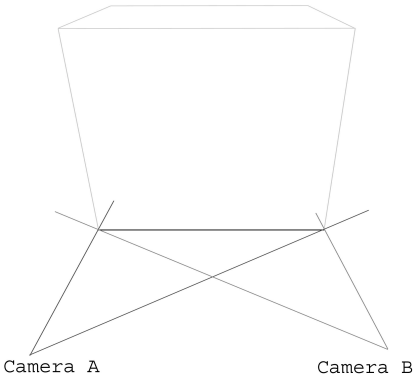

Fig. 5. For horizontal edges near camera level it is difficult to obtain arbitrarily large disparity angles

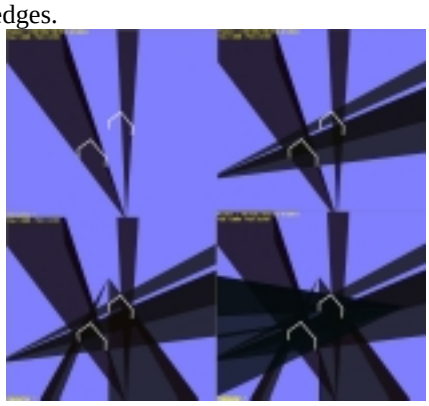

Fig. 6. Projection of primary lines. Primary edges are highlighted in white

Primary edges should be used to recover the core structure of the building, while defining as few edges as possible. Then secondary lines should be used to define all remaining edges. A primary edge must be highlighted in at least three images, although it can be advantageous to define a primary edge in more than three images when trying to recover edges that make poor primary edge candidates. Secondary edges need only be defined in a single image.

2.2 Edge Recovery

After the edges have been highlighted, six automated steps are performed to recover the final edges; Line Projection, Triangle Intersection, Correspondence Recovery, Edge Averaging, Vertex Merging, and Secondary Edge Recovery. Each of these steps is described next.

2.2.1 Line Projection

The first step in determining the positions of the primary edges is to project the 2D primary lines to form 3D triangles. The intrinsic and extrinsic properties of the camera are used to project the primary lines from the cameras position, at the correct orientation out to infinity. This is performed for every primary line in each image, as shown in figure 6 for a scene consisting of 4 images, the final primary edges are highlighted in white.

2.2.2 Triangle Intersection

Once every 2D primary line has been transformed to a 3D triangle, the next step is to determine the intersections between the triangles. Every triangle stores a list of the triangles it intersects.

2.2.3 Correspondence Recovery

Generally each triangle intersects many other triangles even though only a small number of the triangle intersections have both their parent lines highlighting the same edge. Most systems resolve this problem by performing manual correspondences between the lines so that lines which highlight the same edge are grouped together. Once the lines are converted to triangles the only valid intersections are between members of the same group. This can be a very time consuming process. SAMATS performs this correspondence automatically in three steps; Intersection Rating, Triangle Grouping and Group Merging.

2.2.3.1 Intersection Rating

Every triangle needs to rate each of the triangles it intersects. These ratings can then be used to determine which of the intersecting triangles represent the same primary edge as itself. A naïve approach would simply use the coverage of the line of intersection as the only measure in rating each intersecting triangle, with greater coverage resulting in a better rating. This has proved to be almost completely useless because often intersecting triangles which represent a different primary edge (invalid triangles) receive better rating than those that represent the same primary edge (valid triangles).

The automated rating process does not rate an intersecting triangle on the quality of the intersection line, but on the similarity of the intersection line with other intersection lines. This is the reason for having a 3 primary line minimum when highlighting each primary edge.

Figure 7 shows the basis of the rating algorithm in 2D. In the figure there are three cameras, **A B** and **C**, there are two points being modeled, **X** and **Y**, and there are six lines, two from each camera through the points being modeled, $\mathbf{A}_X$ $\mathbf{A}_Y$ $\mathbf{B}_X$ $\mathbf{B}_Y$ $\mathbf{C}_X$ and $\mathbf{C}_Y$. Each line intersects every other line (although some of the intersections are off image) even though the only valid intersections are those between lines with matching subscripts. One should note that the invalid intersections are spaced quite randomly apart while the valid intersection groups have three points of intersection coincident at one location. The automated rating algorithm uses this principle of valid intersections being grouped close together when calculating the rating for each intersecting triangle.

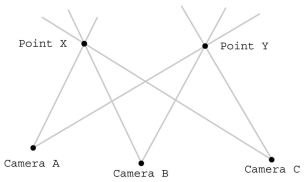

Fig. 7. 2D example of the automated correspondence determination concept

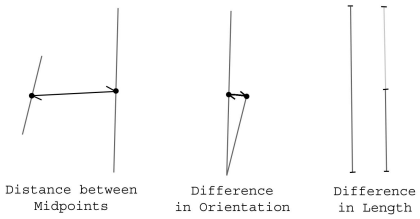

Fig. 8. The three factors considered in determining the similarities between lines of intersection

For each triangle $\mathbf{t}_i$ we need to rate each of the triangles $\mathbf{t}_j$ in $\mathbf{t}_i$'s intersecting triangles set $\mathbf{T}_i$. Since we know that there are at least three triangles per primary edge, we know that at least two of the intersecting triangles are valid matches. We call these valid intersecting triangles $\mathbf{t}_{j1}$ and $\mathbf{t}_{j2}$. Note that there may be more than two valid intersecting triangles, although that fact is not important at this stage. If $\mathbf{t}_{j1}$ is a valid match with $\mathbf{t}_i$ and $\mathbf{t}_{j2}$ is a valid match with $\mathbf{t}_i$, then $\mathbf{t}_{j1}$ and $\mathbf{t}_{j2}$ must be valid matches with each other. This implies that $\mathbf{t}_{j2}$ would be an intersecting triangle of $\mathbf{t}_{j1}$. Therefore, when determining the rating of any $\mathbf{t}_j$, we only need to consider triangles that are in both $\mathbf{t}_i$'s intersecting triangles set and $\mathbf{t}_j$'s intersecting triangles set, i.e. $\mathbf{T}_i \cap \mathbf{T}_j$. Note that only a sub-set of this set will contain valid intersections.

The intersecting triangle $\mathbf{t}_j$ can now be given a ranking based on the triangles in the set $\mathbf{T}_i \cap \mathbf{T}_j$. Each triangle $\mathbf{t}_k$ in this set intersects both $\mathbf{t}_i$ and $\mathbf{t}_j$. Therefore, we can use the three intersecting lines, $\mathbf{l}_{ij}$ $\mathbf{l}_{ik}$ and $\mathbf{l}_{jk}$, to give $\mathbf{t}_k$ a rating. Intersection lines are evaluated based on 3 properties; the distance between their midpoints, the relative orientations between them, and their difference in length.

The value returned for the distance between their midpoints is in the range [0…1] and is described by the following equation;

$$\frac{1}{1+\left(ScalingFactor \times Distance\right)^2}$$

The *ScalingFactor* is used to set the rate at which the value declines with respect to distance. This factor is dependent on the choice of units used to model the building, e.g. if the units are meters and we only want to consider intersection lines with roughly less that 10cm spacing, then setting *ScalingFactor* to about 10 would give a good range. At 1cm the value returned by the equation would be 0.9, at 10cm the value would be 0.5, and at 100cm the value would be 0.09.

The value returned for the measure of the two lines relative orientations is calculated using the absolute value of the dot product between the lines' unit vectors and is also in the range [0…1]. For two lines **A** and **B** the equation is as follows;

$$\left|\hat{A} \cdot \hat{B}\right|$$

Finally, the value returned for their difference in length is in the range [0…1] and is described by the following equation;

$$\frac{\max\left(\overline{A},\overline{B}\right)-\left|\overline{A}-\overline{B}\right|}{\max\left(\overline{A},\overline{B}\right)}$$

Once all these partial tests have been performed, the final rating for the lines is simply the product of the three, which is also in the range [0…1]. Refer to figure 8 for an illustration of each test.

Every triangle $\mathbf{t}_k$ in the set $\mathbf{T}_i \cap \mathbf{T}_j$ is given a rating based on the comparison of the three intersection lines $\mathbf{l}_{ij}$ $\mathbf{l}_{ik}$ and $\mathbf{l}_{jk}$. There are three comparisons that can be made, $\mathbf{l}_{ij}$ with $\mathbf{l}_{ik}$, $\mathbf{l}_{ij}$ with $\mathbf{l}_{jk}$, and $\mathbf{l}_{ik}$ with $\mathbf{l}_{jk}$. The product of these three tests is used to determine the rating of each triangle $\mathbf{t}_k$ in the set $\mathbf{T}_i \cap \mathbf{T}_j$. The product is used in favor of the sum in order to keep the ratings in the range [0…1].

Once every $\mathbf{t}_k$ has been given a rating there are three logical options for assigning a rating to $\mathbf{t}_j$. Assign $\mathbf{t}_j$ the weighted sum of all the ratings in the set $\mathbf{T}_i \cap \mathbf{T}_j$. This has proved unfavorable since this would include triangles that are invalid. If there are a large number of low scoring invalid triangles in a particular $\mathbf{T}_i \cap \mathbf{T}_j$ set, the $\mathbf{t}_j$ will be given a poor rating even if it is a valid triangle.

A second option would be to assign $\mathbf{t}_j$ the weighted product of the ratings. This is a poor choice for the same reasons as assigning the weighted sum, only the problem is amplified greatly when taking the product since there are almost always a few poor scoring invalid matches, this forces the $\mathbf{t}_j$ rating to zero, making the rating useless.

The option that was found to work best is to assign $\mathbf{t}_j$ the best rated $\mathbf{t}_k$ in the set $\mathbf{T}_i \cap \mathbf{T}_j$. If $\mathbf{t}_j$ is a valid intersecting triangle for $\mathbf{t}_i$, the best rated triangle is almost always a valid intersecting triangle for both $\mathbf{t}_i$ and $\mathbf{t}_j$, which we'll refer to as $\mathbf{t}_k$ from here on. When storing the rating for each $\mathbf{t}_j$, a reference to the $\mathbf{t}_k$ triangle responsible for this rating is also stored. This triangle is required for the triangle grouping step described briefly next.

2.2.3.2 Triangle Grouping

After the intersection rating step, for every triangle $\mathbf{t}_i$, every triangle $\mathbf{t}_j$ in $\mathbf{t}_i$'s intersecting triangles set $\mathbf{T}_i$ will have a rating assigned to it. Also, the $\mathbf{t}_k$ responsible for each $\mathbf{t}_j$'s rating will be stored along with the rating. This information can then be used to group triangles together where each group represents a primary edge.

Essentially, the grouping process is performed in two steps. Firstly, the GSS (Group Scope Set) of each triangle is determined. The GSS for each triangle is the list of mutually high ranking intersecting triangles. Not every triangle will have the same size GSS. The size of these sets will vary depending on the number of triangles used to represent each primary edge as well as the relationship between their line intersections.

The second step in the grouping process is to use the GSSs to group the triangles into groups. The triangles are ordered based on the size of their GSS's in ascending order. Triangles with small GSSs form the initial groups. Small GSSs are more tightly coupled which is a desirable property when trying to match triangles together.

After the core set of groups is created all remaining triangles are assigned a group, the vast majority being assigned to one of the existing groups with only a small minority forming their own groups.

It may not be possible to assign every triangle to a group for a number of reasons. The user may not have used three primary lines to highlight a particular primary edge or there may be too great an error to group some primary lines together either due to an error in the camera's intrinsic and/or extrinsic properties or an error in line placement. In such cases the triangles are marked as invalid. For a more detailed explanation of the Triangle Grouping step refer to [4]

2.2.3.3 Group Merging

The final step in the grouping process is group merging. If a primary edge is represented by 6 or more primary lines it may form 2 distinct groups. If the groups were left the way they were, there would be 2 primary edges representing the same building edge instead of just one. The merging step simply compares each group to each other and merges groups which are sufficiently similar.

2.2.4 Edge Averaging

Once all triangles have been assigned a group the primary edges must be determined for each group. This is simply the weighted average of all the intersection lines between all group members.

2.2.5 Vertex Merging

During the edge averaging step, each primary edge will be created totally independently from all other primary edges. In most cases this is acceptable since the majority of primary edges are not connected to any other primary edge. Sometimes however primary edges are connected. This is indicated in the edge highlighting step by having two or more primary lines share the same endpoint.

All primary edges that are connected need to have their connected endpoints coincident. This is achieved by creating a mapping between every primary line and every primary edge, and also between every primary line endpoint and every primary edge vertex. Once the mappings have been made, we can see if any of the primary lines share the same endpoints, which maps to primary edges sharing the same vertex. Once the vertices are identified they are set to the average of their positions.

2.2.6 Secondary Edge Recovery

Secondary edges are determined using the same mapping information obtained during the vertex merging step. Firstly, the secondary lines' endpoints are determined. Then the corresponding vertices are determined for these endpoints and a new group is created for each secondary line using these vertices as the secondary edge's endpoints. After all secondary edges have been highlighted the outline of the model should be complete.

2.3 Structure Recovery

Even though the outline of the model has been determined there is still no surface data associated with the model. The model is only defined in terms of vertices and lines and not in terms of surfaces and the triangles that make up each surface. Recovering this structural information is broken into three steps. The first step is to determine the models surfaces. This is achieved be treating the model like a graph, with the vertices as the graph nodes and the edges as the graph edges. Surfaces are determined by finding the shortest cycles in the graph where all the vertices are co-planar. All surface normals must then be aligned so that they all point away from the model. This is performed by aligning the normals of neighboring surfaces recursively until all normals are aligned. The final step is to triangulate all the surfaces. The algorithm used to triangulate each surface can be found in [8]. Refer to [4] for further details.

3 Texture Extraction

Coming into this section, we have an accurate model of the building, or to be exact, we have a geometrically accurate model of the building. There is still data contained in the image set that has not yet been used to increase the models realism, the buildings façades. The texture extraction process takes the façades from the images and applies them to the model. An overview of this component is presented next. For a more detailed explanation of the Texture Extraction component refer to [4].

3.1 Overview

The aim of the texture extraction process is to produce a 3D model with photorealistic textures. The texture extraction process can be broken into a number of steps. Firstly, the number of images that will contribute to each triangle is determined using back-face culling. There can be any number of contributing images, with each image's contribution first being stored in a temporary texture before they are all blended together per-pixel based on the camera-surface distance and orientation. Occlusion maps are used to prevent incorrect façade data being stored with each triangle. All triangles are then packed into a single large texture retaining the relative size of each triangle, thus creating an authalic texture map. The texture coordinates for each triangle are then set to sample the correct region of the texture map, with the texture then being assigned to the model.

4 Conclusions

This research shows that given sufficient information, user input to the modeling process can be reduced significantly. Currently user input is required for the edge highlighting step but since no correspondence is required this step could be automated

using edge detection and a set of heuristics to guide the choice between using primary lines or secondary lines.

Currently SAMATS has only been used on synthetic images where the exact extrinsic and intrinsic properties of the camera are known. Achieving such precision in the real world would prove difficult without specialized equipment. New techniques will be required to facilitate the gathering of the geo-referenced images required by SAMATS in order for this system to be utilized effectively in the real world.

SAMATS has shown the ability to model rectangular and triangular roofed structures very well, however SAMATS does have trouble modeling certain structures. SAMATS has no special ability to handle curved surfaces, which makes it impossible to model such features completely accurately. Cylindrical column must be replaced by rectangular columns for instance. Another difficulty that can arise is SAMATS' inability to handle partially highlighted edges. This makes it difficult, and in some cases impossible, to model buildings in tightly confined spaces.

References

1. J.D. Carswell, A. Eustace, K. Gardiner, E. Kilfeather. An Environment for Mobile Context-Based Hypermedia Retrieval, in 13th International Conference on Database and Expert Systems Applications (DEXA2002); IEEE CS Press; Aix en Provence, France; September 2002
2. S.R. Coorg. Pose Imagery and Automated Three-Dimensional Modeling of Urban Environments. PhD thesis, MIT Ph.D. Thesis, 1998.
3. P.E. Debevec, C.J. Taylor, and J.Malik. Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach. In SIGGRAPH '96 Conference Proceedings, 11-20, 1996.
4. J. Hegarty and J. Carswell. SAMATS – Semi-Automated Modeling And Texturing System. Masters Thesis, 2005.
5. S.C. Lee, S.K. Jung, and R. Nevatia. Integrating Ground and Aerial Views for Urban Site Modeling. In Proceedings of International Conference on Pattern Recognition, 2002.
6. S.C. Lee, S.K. Jung, and R. Nevatia. Automatic Integration of Façade Textures into 3D Building Models with a Projective Geometry Based Line Clustering. Computer Graphics Forum, 21(3):511-519, 2002.
7. S.C. Lee, S.K. Jung, and R. Nevatia. Automatic Pose Estimation of Complex 3D Building Models. Proceeding of the 6th IEEE Workshop on Applications of Computer Vision, 2002.
8. J. O'Rourke. Computational Geometry in C (Second Ed.). Cambridge University Press, 1998.
9. C.J. Taylor and D.J. Kriegman. Structure and Motion from Line Segments in Multiple Images. PAMI, 17(11):1021-1032, November 1995.
10. S.Ullman. The Interpretation of Structure from Motion. Proceedings of the Royal Society of London, 1976.
11. S. Zlatanova and F.A. van den Heuvel. Knowledge-based Automatic 3D Line Extraction from close range images. Web – http://www.gdmc.nl/zlatanova/thesis/html/refer/ps/SZ_FH_Corfu.pdf