ISPRS Journal of Photogrammetry & Remote Sensing 54 (1999) 342–351

PHOTOGRAMMETRY
& REMOTE SENSING

Development of polygon shift method for generating 3D view map of buildings

Shunji Murai [a,*], Vivarad Phonekeo [a], Kunihiko Ono [b], Shigetaka Yasue [b]

[a] Space Technology Applications and Research Program (STAR), Asian Institute of Technology, PO Box 4, Khlong Luang, Pathumthani 12120, Thailand

[b] Geoinfo Creative, 2-4 Kanda Jinbocho, Chiyoda-ku, Tokyo 101, Japan

Received 18 June 1998; accepted 23 February 1999

Abstract

The authors have developed a new method termed ''Polygon Shift Method'' that enables the generation of a 3D view map of a city with tall buildings with a simplified procedure to shift a polygon and check the overlap between the original and shifted polygon. Boolean operations are applied with a newly defined ''Fore or Aft'' side and a ''Depth Distance'' that functionally express the visibility criteria or hidden point processing in the 3D view. Since the polygon shift method can be operated with a raster-based structure, the computer processing for generation of a 3D view map of buildings with shadow is simple and efficient. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: 3D view map; polygon shift method; shadow analysis; hidden point algorithm

1. Introduction

The perspective of buildings or landscape used to be drawn manually by architects. Since computer-aided design (CAD) was introduced in the 1970s, automated drawing became popular (Murai, 1997; Wilson, 1998). In the recent decade, virtual reality software is available to produce perspectives or bird-eye views of 3D structures. However, data format and structure are not yet very simple, because all details, not only of the building plan, but also nodes, edges and surfaces (vertical walls and roofs) should be input with specified topology. Various data structures of 3D GIS have been proposed by e.g., Molenaar (1990), Shibasaki and Huan (1992), Chen and Ikeda (1994), Tempfli (1997) and William et al. (1997). All these data structures are based on vector topology.

The polygon shift method proposed in this paper needs a simple data structure of a polygon (building plan given by vector data as the input data) to generate a raster belonging to the polygon. Data about roof and vertical walls are not necessary because the method only targets prismatic shaped buildings with horizontal roofs, although any compli-

* Corresponding author. Murai Laboratory Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro, Tokyo, 153-8505 Japan. Tel.: +81-3-5452-6407; Fax: +81-3-5452-6408; E-mail: murai@shunji.iis.u-tokyo.ac.jp

PII: S0924-2716(99)00031-3

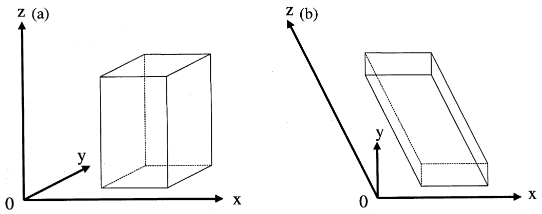

Fig. 1. Proposed 3D coordinate system for the 3D view map. (a) Conventional 3D coordinate system with vertical z-axis and oblique xy-axes. (b) Proposed 3D coordinate system with oblique z-axis and orthogonal xy axes.

cated building can be formed as a combination of multiple columns. Using vector data, the geometry to achieve hidden point/line/surface processing for multiple buildings needs a very complicated algorithm, when the building shape includes concave parts. For example, the intersection of lines and surfaces and the identification of overlap areas of two polygons will be a complicated calculation for concave polygons. A comparison between the proposed method and some existing algorithms (Z-buffer, scan-line and back-face removal) is described in Section 5.

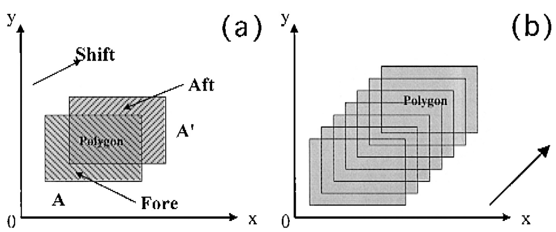

Fig. 2. Concept of polygon shift method. (a) Definition of polygon shift. (b) Number of polygon shifts.

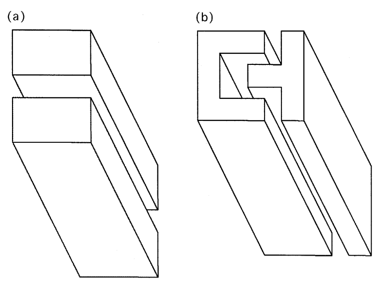

Fig. 3. Hiding and hidden buildings. (a) A simple case with convex-shaped buildings (closer building hides part of a farther building). (b) A complicated case with a concave-shaped building (it is difficult to judge which building is closer or farther).

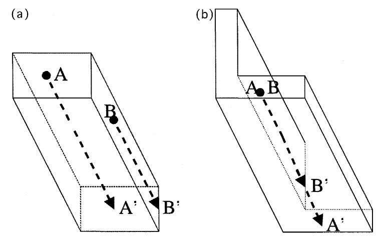

Fig. 4. Concept of depth distance. (a) Depth distance of roof (point A) and vertical wall (point B) points. (b) Comparison of depth distance between overlapping ''Fore'' sides; A on the roof and B on the vertical wall (if $AA' > BB'$, then A hides.

In this study, a raster-based approach with Boolean operations works efficiently to identify overlapping areas using the depth distance, which is a criterion to judge which part of a building hides other buildings.

2. Proposed 3D coordinate system

In the conventional method, most of the 3D view maps used to be based on x, y, z coordinates with a

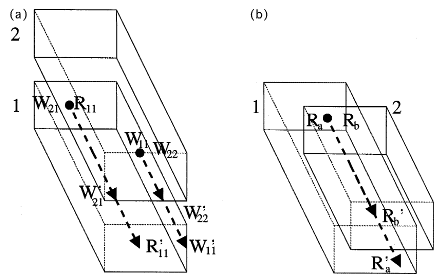

Fig. 5. Hiding and hidden parts using the depth distance. (a) Overlap between roof and wall (points R_{11} and W_{21}) and between walls (ponits W_{11} and W_{22}). (b) Overlap between roofs.

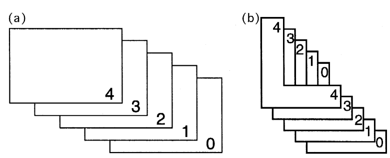

Fig. 6. Depth distance in case of polygon shift. (a) Depth distance of a convex shaped building. (b) Depth distance of a concave-shaped building.

horizontal *x*-axis, an oblique *y*-axis and a vertical axis, as shown in Fig. 1(a). In this study, we use a different 3D coordinate system, called Military Projection, with an orthogonal *xy* coordinate system and an oblique *z*-axis, as shown in Fig. 1(b). The benefit of this system is that we do not need to change the original shape of the building plan, while the disadvantage is that the view map looks a bit strange with oblique buildings.

3. Concept of polygon shift

The polygon shift can be defined as shift of polygons to a certain direction termed as ‘‘shift direction’’ with a small distance termed ‘‘shift amount’’, as shown in Fig. 2(a). Then, the original polygon A and the shifted polygon A′ will create three parts by Boolean operations:

1. Overlapping area: A and A'
2. A part of A not included in A': $A-(A$ AND $A')$
3. A part of A' not included in A: $A'-(A$ AND $A')$

In this study, the area defined by (2) is called ‘‘Fore’’ while the area defined by (3) ‘‘Aft’’. Visible parts of vertical walls will be identified by ‘‘Fore’’, while invisible parts will be detected by ‘‘Aft’’.

The 3D view of a building with the roof and visible vertical walls can be generated by a sequence of polygon shifts with the shift repetition according

Table 1
Comparison between existing methods and the proposed method

Existing methods	Polygon shift method (Fig. 7(d))
Z-buffer algorithm (Fig. 7(a)); There are two algorithms, Z-buffering and Z-sorting. The Z-buffering algorithm determines which parts are visible or hidden by comparing *Z*-axis values of each pixel in 3D graphics. With Z-buffering, the graphics processor stores the *Z*-axis value of each pixel in a special area of memory called the Z-buffer. The maximum *Z*-axis value corresponds to the visible pixel. Z-sorting, an alternative algorithm simply displays all objects (buildings in this study) serially, starting with those objects farthest back with the largest *Z*-axis values. The Z-sorting algorithm does not require a Z-buffer, but it is slow and does not render intersecting objects correctly. Both algorithms need geometric calculations to obtain *Z*-axis values of a pixel, which may be located on different faces of the same building and/or different building. The process is required for all possible combinations of overlapping buildings or faces.	The indicator to determine hiding and hidden parts is called ‘‘depth distance’’ and is similar to the *Z*-axis value in the Z-buffer algorithm. However, the depth distance in the polygon shift method is not necessary to calculate like in the Z-buffer algorithm. Depth distance is given automatically in the process of the sequential polygon shift, where the building plans are polygons represented in raster mode and the polygons are sequentially shifted at the given interval in the looking angle. At each shift repetition, the depth distances are incremented by 1 starting with 0 values. The longest depth distance of each pixel between the overlapping parts corresponds to the visible pixel in a full-screen buffer. No geometric calculation is required but only logical analysis (longer or shorter depth distance).
Scan-Line algorithm (Fig. 7(b)); Along a scan-line, which traverses given buildings in 3D projection, the *Z*-axis values are computed and the maximum value of each pixel along the scan-line are kept in memory as visible parts. A full-screen Z-buffer is not needed, if we consider all polygons for a scan-line before moving to the next scan-line.	The disadvantage of the polygon shift method is the need of a full-screen memory, while the advantage is that there is no need to identify on which faces of what building a pixel is located and to calculate the *Z*-axis value.
Back-face removal algorithm (Fig. 7(c)); All back-faces are removed by using the normal vector to the plane of the polygon to identify which faces are front or back. A 3D view can be generated efficiently, but one cannot identify which faces are hidden by other faces.	In the polygon shift, all invisible parts are automatically removed by using the topology of ‘‘Aft’’ defined in Section 3. It can be identified which parts did the visible parts with the maximum depth distance hide, if all shorter depth distances were kept in memory, when the comparison was made.

to the height of the building, as shown in Fig. 2(b). In case of prismatic type buildings, to which we are limited in this study, the roof represents the finally shifted polygon, while the visible walls represent a series of ‘‘Fore’’ parts. The invisible walls are automatically hidden in ‘‘Aft’’ parts in the shift process.

4. Determination of hiding and hidden parts

There would be no problem for determination of hiding and hidden parts, if 3D view images of buildings were displayed on a monitor sequentially in the order from far to close because far buildings would be automatically hidden. This is true only if these buildings have a convex shape as the simple case shown in Fig. 3(a). However, if there are concave shape buildings as shown in Fig. 3(b), it is complicated to judge which building is farther or closer than the other one. Therefore, there should be a theoretical criterion to judge the visibility.

In this study, a distance from a visible point on a roof or wall to the base termed ‘‘Depth Distance’’ (see Fig. 4(a)) is used for the determination of hiding

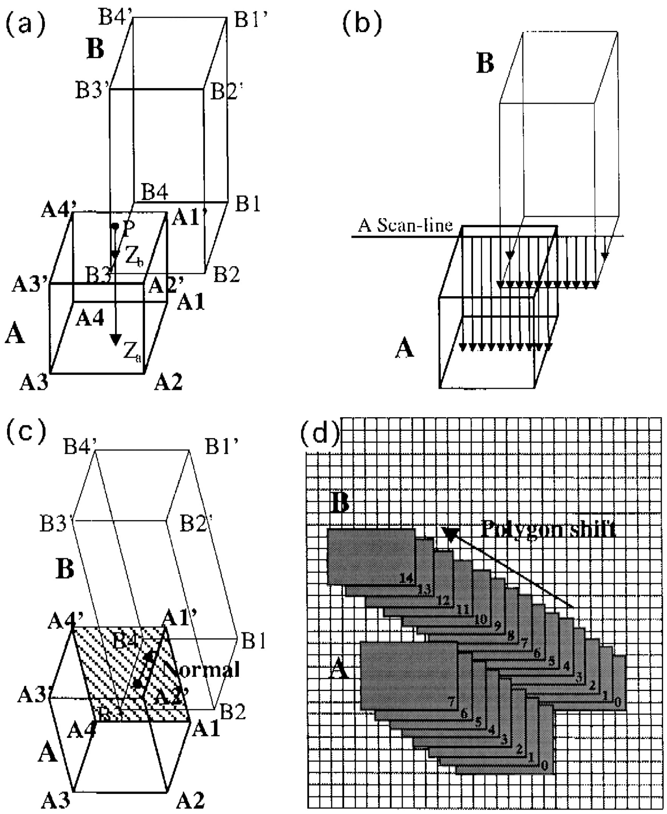

Fig. 7. Schematic comparison between existing methods and the proposed method. (a) Z-buffer algorithm. The Z value of ponit P on the face A'_1 A'_2 A'_3 A'_4 (roof A) of building A, Z_a, is bigger than the Z value of point P on the face B_4 B_3 B'_3 B'_4 of building B, Z_b. Thus, the point P is located should be searched and the Z values should be calculated. A full-screen Z-buffer should be kept in memory. (b) Scan-line algorithm. instead of the pixel-based analysis using a full-screen Z-buffer, the determination of hiding and hidden parts is implemented scan-by scan-line for all buildings. however, a similar problem to calculate Z-values remains as in the Z-Buffer algorithm. (c) Back-face removal algorithm. The back-face $A_1 A_4 A'_4 A'_1$ of building A can be removed bu using the normal vector to the plane to identify whether the face is in front or back. But it cannot be identified which faces hide other ones. (d) Polygon Shift Method. polygons are shifted layer by layer at the interval of a unit polygon shift, when the depth distance is incremented depending on the amount of the polygon shift. The pixels with the maximum depth distance are kept in a full-screen memory as the visible parts.

and hidden parts. If there are overlapping areas among ''Fore'' sides (see example in Fig. 4(b)), a point with a longer ''Depth Distance'' hides another point with a smaller ''Depth Distance''. In the shown example, point A on the roof hides point B on the wall, which is originally assigned ''Fore'' visible side according to the definition given in Section 3.

In case of two or more buildings, as shown in Fig. 5, the depth distance of a point on overlapping polygons between roof and wall, between walls (see Fig. 5(a)) and between roofs (see Fig. 5(b)) allows the determination of hiding and hidden parts. Points R11 of building 1 and W11 of building 2 in Fig. 5(a) and Ra of building 1 in Fig. 5(b) hide points W21 and W22 in Fig. 5(a) and Rb in Fig. 5(b), respectively of building 2.

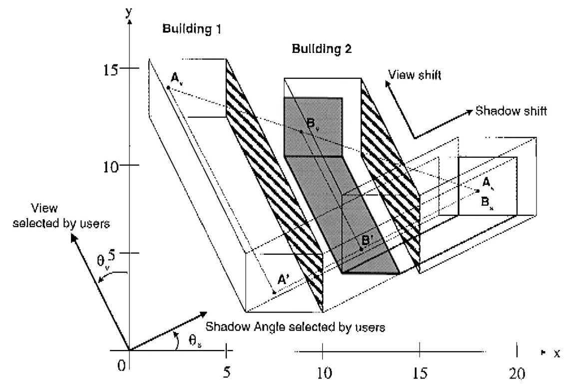

Fig. 8. Shadow analysis using the depth distance. point A_v of Building 1 shadows Point B_v of Building 2 as analysed using the '' Depth Distance'' ($A_s A' > B_s B'$).

5. Comparison between the polygon shift method and existing algorithms

5.1. Algorithm of the polygon shift method

The polygon shift method is composed of the following steps:

Step 1: input building plan in vector form and the building height.

Step 2: set up a raster of the study area with the resolution as specified by the user and convert the input building plans into raster-based polygons.

Step 3: input shift direction as a function of the looking angle and the depression angle, and shift amount as a function of the building height.

Step 4: clear the buffer of the depth distance of each polygon as an initial value.

Step 5: shift each polygon with a spacing of the unit shift amount in the given shift direction and add 1 to each shifted polygon as the depth distance.

Step 6: compare the value of the former buffer with the one of the shifted buffer at each pixel of the polygons and select the longer depth distance.

Step 7: repeat steps 5 and 6 until all polygons are shifted with respect to the shift amount given as function of the height.

Step 8: output the depth distance of all pixels.

Step 9: assign grey scale or colour in consideration of the shading and the shadow effect.

In case of a convex-shaped building as shown in Fig. 6(a), the polygon shift method results in a visible roof (the shifted polygon) and visible walls (the repeated ''Fore'' parts) as defined in Section 3. All ''Fore'' parts are visible in case of a convex-shaped building, while some of ''Fore'' parts are invisible in case of a concave-shaped building as shown in Fig. 6(b). However, in the polygon shift method, we do not need to classify whether a given building is convex or concave. The steps 5 and 6 implement automatically the determination of hiding and hidden parts regardless of a convex or concave building. In case of the determination of hiding and

Table 2
Visibility check and image output

Case	View shift	Shadow shift	Image output
1	Fore	Fore	Paint dark only shadowed areas
2	Fore	Aft	Paint dark all areas
3	Aft	Fore	Neglect
4	Aft	Aft	Neglect

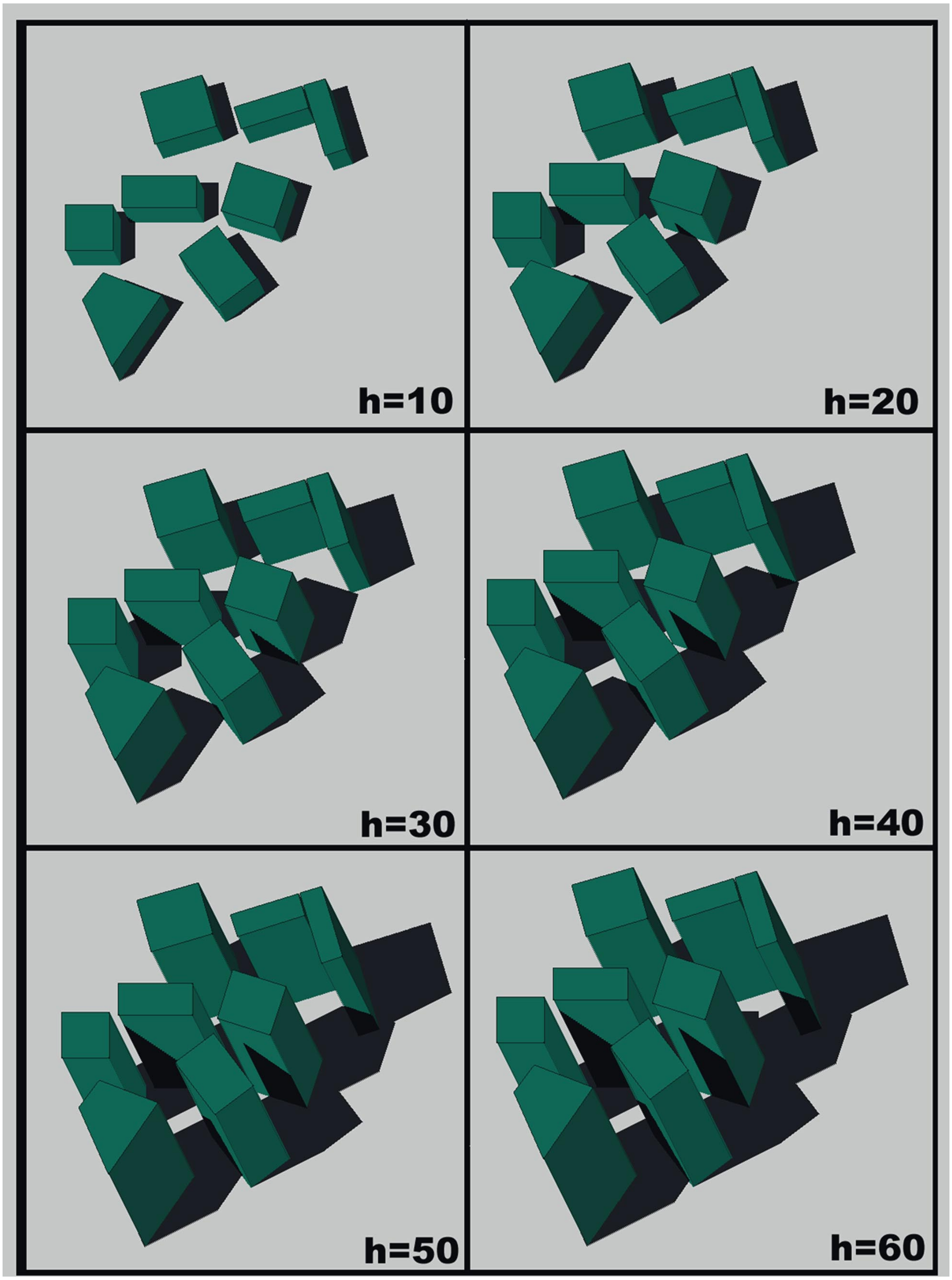

Fig. 9. Case studies for generating 3D view maps with shadow using the polygon shift method for different building heights (h) in pixels.

Table 3
Computation times (in s) with respect to the height H (in pixels) and the sub-programmes

Sub-programme	Case 1 $H=10$	Case 2 $H=20$	Case 3 $H=30$	Case 4 $H=40$	Case 5 $H=50$	Case 6 $H=60$
View shift	11.42	14.35	16.69	21.77	25.86	29.51
Shadow shift and integration	401.96	423.30	443.43	469.93	497.16	540.71
Image output and other	31.87	36.84	38.84	45.06	50.17	56.43
Total	445.25	474.49	498.96	536.76	573.20	626.65

hidden parts between multiple buildings, the same algorithm of steps 5 and 6 in the repeated process with respect to the heights of multiple buildings solves automatically the visibility problem.

The biggest advantage of the polygon shift method is that we do not need to identify one surface and another within a building as well as between different buildings to determine the hiding and hidden parts, as is usually the case in existing algorithms. The process of the polygon shift with combination of the polygon shift operations and the comparison of the depth distances of each pixel result in a 3D view map with only visible roofs and walls.

5.2. Comparison with existing algorithms

Three algorithms, Z-buffer, Scan-Line and Back-Face Removal algorithm, were selected for a comparative study on the determination of hiding and hidden parts (Harrington, 1987; Z-buffer, 1999).

As an indicator of the determination of hiding and hidden parts, the Z-buffer algorithm uses the Z-axis value of each pixel. The idea of the Z-Buffer is similar to the depth distance in the polygon shift. The difference is that the procedure of the Z-buffer algorithm, both Z-buffer and Z-storing, has to be implemented for all possible combinations of a pair of two different surfaces (roofs and walls) using pixel-based analysis. In this algorithm, it is required to identify on which faces of what building a pixel to be compared is located in a 3D projection plane. This makes the Z-buffer algorithm more complicated than the polygon shift method.

The scan-line algorithm is basically similar to Z-buffer algorithm except that it uses a scan-line instead of a pixel-based approach in order to avoid a full-screen buffer memory. Therefore, it has basically the same disadvantages.

The back-face removal algorithm is based on identifying surfaces which are front or back and on removing the back-faces by finding the normal vector to the plane of polygon. This algorithm seems very simple, but one cannot identify which surfaces hide other surfaces. Therefore, it is difficult to expand this algorithm to shadow analysis.

One of the benefits of the polygon shift is that it does not need to give the wall surfaces as input data. Wall surfaces will be automatically generated in the process of the polygon shift together with visibility check by using the concepts of ''Fore'' and ''Aft'' as well as ''Depth Distance''. Roof surfaces are always the same in shape as the input polygon, but shifted in location, because in this study only prismatic type buildings are considered.

Table 1 shows the summary of the comparison between existing methods and the polygon shift method. Fig. 7 shows schematically the Z-buffer, scan-line, back-face removal algorithm and polygon

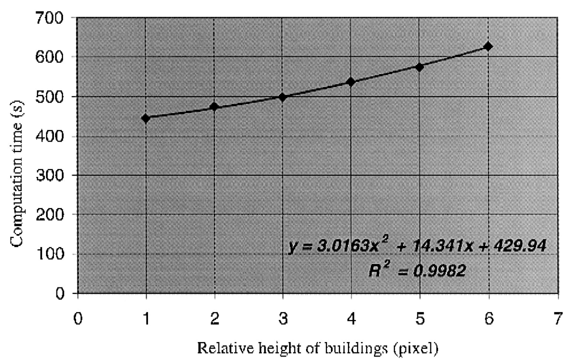

Fig. 10. Computation times in relation to the height of the given buildings (see Fig. 9). The quadratic equation is valid for the specific computer configuration and 7 buildings all with the same height for all 6 cases (height from 10 to 60 pixels). The relative heights x are the real heights divided by the minimum height, i.e., express the relation between the heights.

shift method and the differences among them. In these three existing methods, even those invisible faces, that are automatically avoided in the Polygon Shift Method by using the concept of ''Aft'', should be searched and checked for visibility using complicated geometry. The disadvantage of the polygon shift method is that it needs a full-screen ''Depth Distance'' buffer as the Z-buffer algorithm. The required memory depends on the resolution or pixel spacing to represent the input polygons and the shift direction and the total shift amount given by the height of building. Another disadvantage is that the

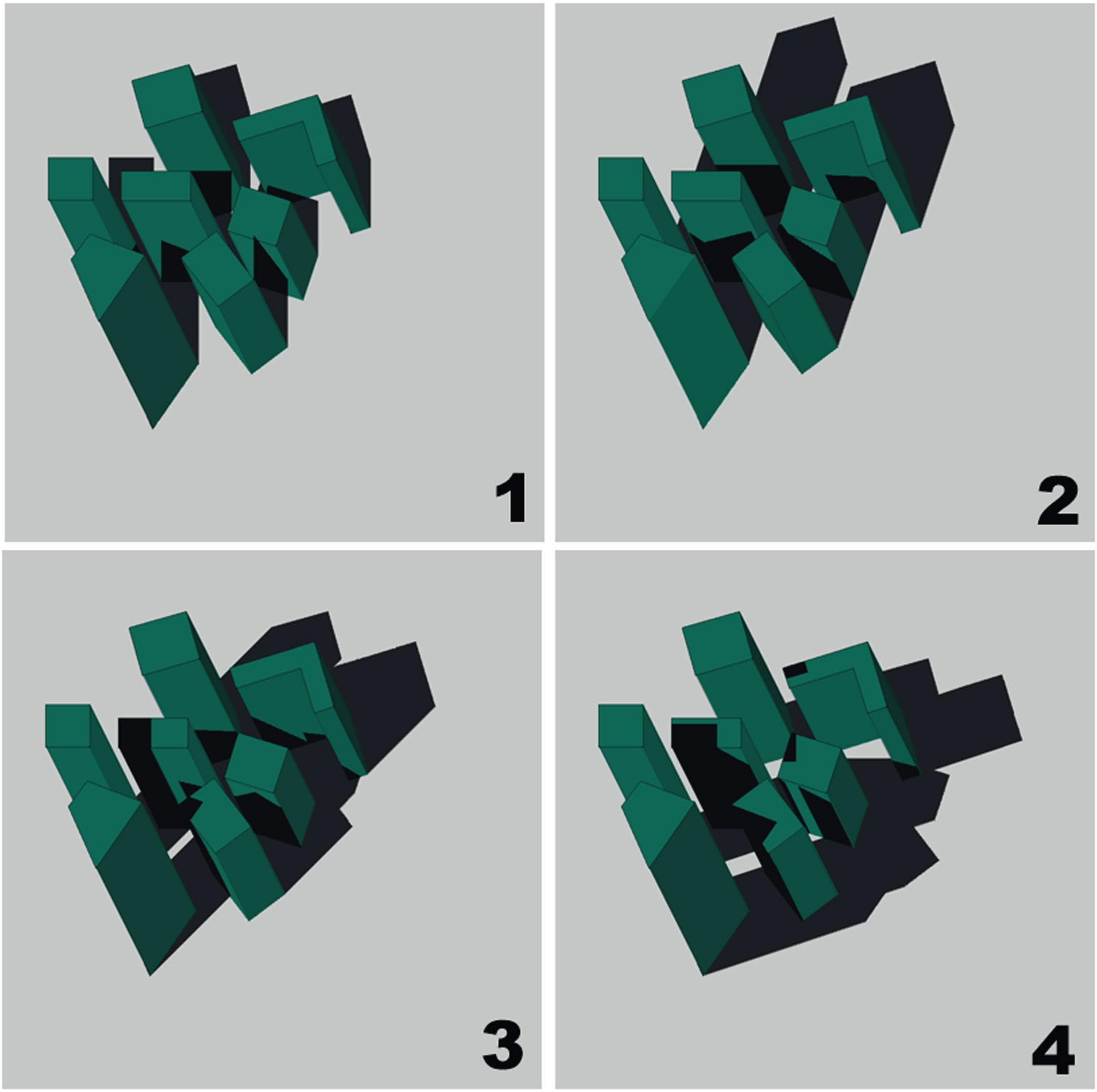

Fig. 11. Various examples of 3D view map of buildings with different height and view/shadow shift direction.

shape of buildings will not be very precise if the pixel resolution (density) is low. However, there would not be big memory problems with recent computers, including PCs.

6. Shadow analysis with depth distance

Shadow is created by specifying a shift direction as an illumination direction, a shift amount and the shift repetition (shadow length), similarly to the case of the 3D view map. In this paper, the polygon shift for 3D view map generation is termed ‘‘View Shift’’, while for the shadow analysis is termed ‘‘Shadow Shift’’.

The polygon shift approach for both cases is exactly the same for checking the visibility of overlapping areas using the depth distance, though ‘‘hiding and hidden’’ analysis in the 3D view map is replaced by ‘‘shadowing and shadowed’’ analysis in the second case. This visibility check will be implemented only to ‘‘Fore’’ sides in both view and shadow shifts. There are four combinations with respect to ‘‘Fore’’ and ‘‘Aft’’, and ‘‘View’’ and ‘‘Shadow’’ shifts as shown in Table 2. As the output image for the 3D view of buildings with shadow will be made in the ‘‘View Shift’’ space, only ‘‘Fore’’ sides in the ‘‘View Shift’’ are taken into account.

The concept of shadow analysis using the depth distance only in ‘‘Fore’’ sides of view and shadow shift spaces is shown in Fig. 8. The hatched areas show the case 2 with ‘‘Fore’’ view shift and ‘‘Aft’’ shadow shift.

The computation time required for the integration of the ‘‘View Shift’’ and ‘‘Shadow Shift’’ depends on the height of building, the size of shift amount and the resolution of pixel spacing. As the shift amount and the resolution affect the computation time as a linear and quadratic function respectively, an experimental study was implemented to test the computation time with respect to the change of height.

In order to simplify the experiment, seven buildings were given the same height in each case. Six cases with different height ranging from 10 to 60 pixels in the shift amount were implemented with a Pentium of 200 MHz and 64 MB memory. A full-screen memory of 700×700 pixels $\times$ 3 colour bands (1.47 MB) is used for the ‘‘depth distance’’ buffer in both ‘‘View’’ and ‘‘Shadow’’ shift. Fig. 9 shows the six case studies for generating 3D view maps with shadow using the polygon shift method.

Table 3 shows the computation times with respect to the change of height and the sub-programmes. As seen in the table, about 90% of the computation time was required for the shadow shift analysis and the integration with the view shift analysis because two full-screen buffers are to be checked. The total computation time can be represented as a quadratic function with respect to the height as shown in Fig. 10. When the polygon shift method is applied only to the generation of a 3D view map without shadow, the computation time is less than 90 s for these six case studies. Fig. 11 shows four examples of a 3D view map with different shadow shift for the same building plan that was used in the above case studies.

7. Conclusions and further studies

A polygon shift method was proposed and demonstrated as a powerful algorithm to generate 3D views of buildings with shadow without any complicated data structure. The concept of polygon shift, including shift direction, shift amount, shift repetition, fore and aft polygon regions and depth distance, was recognised to simplify the geometric, topologic and logical operations required for 3D visualisation. It can overcome the weakness of raster data structure with respect to topology by introducing the above concepts. The Polygon Shift Method can be widely applied to other 3D objects, for example, topography represented by contour lines.

Further studies should be made to extend the polygon shift method to inclined roofs and perspective views. An improvement to reduce the computation time required for the shadow shift analysis and the integration with the view shift analysis has to be achieved.

References

Chen, X., Ikeda, K., 1994. Three Dimensional Modelling of GIS Based on Delaunay Tetrahedral Tessellations. Proc. ISPRS Comm. III Symposium, Munich, Germany. In: Int’l Arch. Photog. and Rem. Sens., Vol. XXX, Part B3/1, pp. 124–131.

Harrington, S., 1987. Computer Graphics, A Programming Approach. McGraw-Hill Book, Singapore, pp. 312, 319–320.

Molenaar, M., 1990. A formal data structure for three dimensional GIS in geographic information systems. Proc. of 4th International Symposium on Spatial Data Handling, Vol. 2 pp. 830–843.

Murai, S., 1997. GIS Work Book. Technical Course, Japan Association of Surveyors, January, pp. 141–149.

Shibasaki, R., Huan, S., 1992. A digital urban space model — a three dimensional modelling technique of urban space in a GIS environment. Proc. ISPRS XVIII Congress, Washington DC, USA. In: Int'l Arch. Photog. and Rem. Sens., Vol. XXIX, Part B4, pp. 257–264.

Tempfli, K., 1997. Photogrammetry for Urban 3-D GIS. GIM International 11 (1), 60–63.

William, D., Goldsmith, V., Toban, J., Beeker, M., 1997. Urban topography provided new challenges in 3-D modelling. GIS World 10 (9), 64–68.

Wilson, J.D., 1998. Shopping Around for Data. GIS Asia Pacific, December 1997/January 1998, pp. 24–28.

Z-buffer, 1999. Webopaedia Definition and Links. http://webopedia.internet.com/TERM/Z/Z_buffer.html, accessed 7 October, 1999.