TRIANGLE-BASED VISIBILITY ANALYSIS AND TRUE ORTHOIMAGE GENERATION

KAZUO ODA WEI LU OSAMU UCHIDA TAKESHI DOIHARA
ASIA AIR SURVEY CO., LTD.

Commission III, PS WG III/8

KEY WORDS: Orthorectification, Photogrammetry, Algorithms, Orthoimage, Transformation, Visibility Analysis, Triangular Prism Model, 3-D space description

ABSTRACT:
We propose a new method that can realize rapid generation of true orthoimages from area-type sensor images, accelerated by triangle-based visibility analysis. A Triangular-Prism Model (TPM) for 3-D space description and triangle-based visibility analysis is introduced. TPM, considered as an extension of TIN model, consists of triangular prisms and every top triangle is the surface of the ground, buildings, and other objects on the ground. Vertical walls also can be expressed with the side faces of prisms.
TPM simplifies visibility analysis between elements of surface model. The visibility among a group of triangular prisms can be related with that of a group of 2-D triangles that are the projection of the prisms onto 2-D space. This paper introduces visibility sorting, where triangular prisms are sorted according to the visibility from the viewpoint.
The visibility sorting has been applied to true orthoimage processing where occlusions by buildings are essential. Tests with 30 aerial images of Shinjuku area shows that occluded area around building can be extracted in each true orthoimage. True orthoimages can be synthesized into one composite true orthoimage where occluded areas of one image are compensated with other true orthoimages.

1. INTRODUCTION

Many of the existing methods of true orthoimage generation are facilitated by separation of DSM into DTM and DBM (Digital Building Model). These types of true-orthoimage generation simplify visibility analysis by counting the cases where some parts of DBM may occlude DTM, and omit the cases of occlusion between DBM and DBM, or DTM and DTM. Other types of methods perform Z-buffering for each pixel of orthoimage and requires large amount of processing time.
Our method adopts Triangular Prism model (vertical faces like building walls are allowed) for DSM and does not distinguish between DTM and DBM. This means that this method has no limitation in the target of visibility analysis. The theory for triangle-based visibility analysis utilizes the fact that triangle planar surfaces in the model never intersect with each other, except the case that triangles contact at their edges or surfaces.
TPM simplifies visibility analysis between elements of surface model. Visibility among a group of triangular prisms can be related with visibility of a group of 2-D triangles that are projection of the prisms onto 2-D space. This paper also introduces visibility sorting, by which triangular prisms are sorted according to the visibility from the viewpoint. Such visibility sorting can be applied to true orthoimage processing where occlusions by buildings are essential.
This paper first introduces the definition of TPM and describes how to create TPM from a digital map. The following section explains visibility analysis of TPM and its base triangles, and introduces visibility sorting. Section 4 treats the procedure for true orthoimage generation based on visibility sorting of TPM. The procedure of orthorectification is demonstrated in Section 5.

2. DEFINITION OF TRIANGULAR PRISM MODEL

2.1 Definition

Triangle Prism Model (TPM) is a group of triangular prisms of finite number, satisfying the following conditions (Figure 1).

(1) All side faces are vertical and vertical projection of the top face and the bottom face are the same. This triangle is called "base triangle".
(2) All prisms have no common parts except for side faces. Similarly, all base triangles have no common parts except for edge line segments.

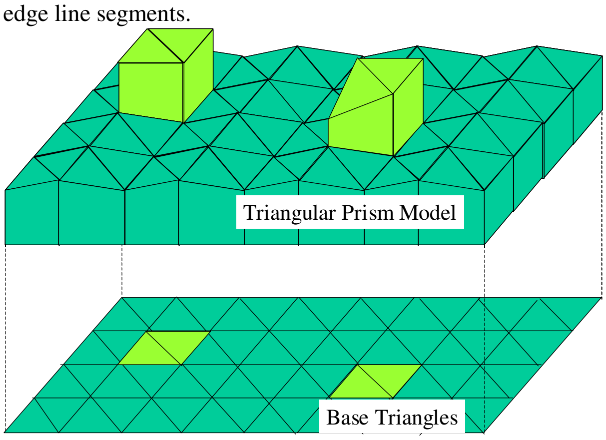


Figure 1. Triangular prism model (TPM) and base triangles.

2.2 TPM Creation from Digital Maps

TPM can be constructed from ordinary 3-D digital maps. Before creation, features in digital maps should be categorized into two types of feature groups: one includes features on the ground, such as roads, vegetations, or elevation contours. The other group includes features above the ground, typically building polygons which have elevation at roofs. Then 2-D TIN is created from all feature points and line segments. Triangle prisms are generated for each triangle in the TIN. Top faces of the prisms for the features on the ground lay on ground surface, while those for the features above the ground are the roof tops. Bottom faces of all prisms may have a certain elevation value that is equal to or lower than the lowest elevation of the features.

3. VISIBILITY ANALYSIS OF TPM

3.1 Visibility Relationship Between Prisms and Base Triangles

In this paper, the following expression is used in this paper for prism A (or triangle A) that is hided by prism B (or triangle B) from 3-D view point O (or 2-D viewpoint O):

$$A \triangleright B : O$$

And if prism A (or triangle A) is not hided by prism B (or triangle B), i.e. $A \triangleright B : O$, the following expression is used in this paper:

$$A \ntriangleright B : O$$

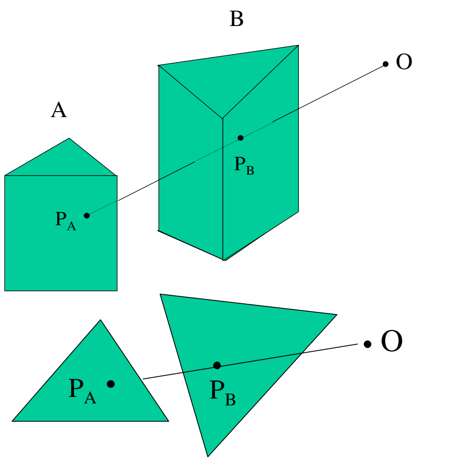


Figure 2. Visibility relationship from viewpoint O.

The following theorem simplifies visibility relationship between triangular prisms in TPM.

Theorem 1: For any pair of triangular prism A, B in TPM, if

$$A \triangleright B : O$$

then

$$B \ntriangleright A : O$$

The following theorem relates the visibility of triangle prisms from the viewpoint to visibility of base triangles.

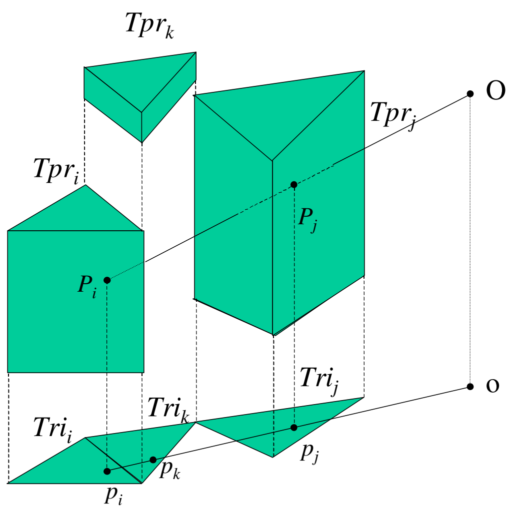


Figure 3. Theorem 2.

Theorem 2: Let Tpr be TPM which includes prisms $\{Tpr_1, Tpr_2, \ldots, Tpr_n\}$, the base triangle group of Tpr be $Tri := \{Tri_1, Tri_2, \ldots, Tri_n\}$, and O be the vertical projection of 3-D viewpoint O, called 2-D viewpoint corresponding to O. For any pair of i and j $(i \neq j)$,

$$Tpr_i \triangleright Tpr_j : O \Rightarrow Tri_i \triangleright Tri_j z : o$$

And as the contrapositive of the upper proposition,

$$Tri_i \ntriangleright Tri_j : o \Rightarrow Tpr_i \ntriangleright Tpr_j : O$$

3.2 Visibility Sorting

3.2.1 Definition: The result of visibility sorting of triangle prisms from 3-D viewpoint O is a sequence $S := \{S_1, S_2, \ldots, S_n\}$ in which any pair of prisms,

$$i < j \Rightarrow S_i \ntriangleright S_j : O$$

Similarly, a visibility sorting sequence of base triangles from 2-D viewpoint o is defined as a sequence $s := \{s_1, s_2, \ldots, s_n\}$ in which any pair of triangles,

$$i < j \Rightarrow s_i \ntriangleright s_j : o$$

Here we call a procedure to obtain a visibility sorting sequence "visibility sorting".

Theorem 2 leads to the following important theorem.

Theorem 3: If $STri$ is a visibility sorting sequence of base triangles of a TPM, the sequence of prisms $STpr$, where prisms corresponding to $STri$ are arranged in the same order, also composes a visibility sorting sequence.

3.2.2 Visibility Sorting of Triangles: Visibility sorting of triangles from 2-D viewpoint O can be executed in the following steps.

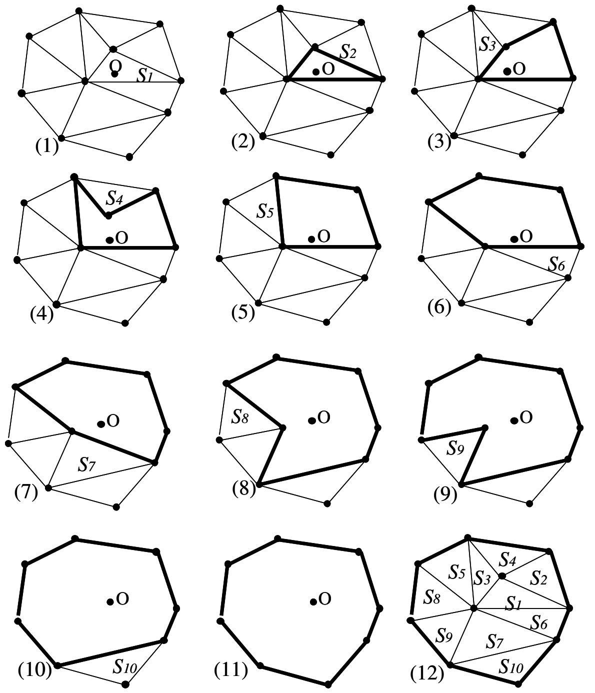
Step 1: Let the triangles be $Tri := \{Tri_1, Tri_2, \ldots, Tri_n\}$. Generate TIN with triangles edges and corners. If the viewpoint o is outside of triangles, include o for TIN generation. Let these triangles be $Tri' = \{Tri'_1, Tri'_2, \ldots, Tri'_{n'}\}$.

Step 2: Choose one triangle which includes o. Let it be the first triangle S_1 in the sorted triangle sequence, and remove it from the TIN (see Figure 4 (1)). We call the polygon around removed area "front polygon". Here the front polygon is S_1 itself.

Step 3: Remove one triangle that is not hidden by other triangles from o. Add the triangle to the sorted triangle sequence.

Step 4: Repeat Step 3 and remove until all triangles in TIN are removed (see Figure 4 (2)-(12)).

Step 5: From the sorted triangle sequence, remove triangles not included in $Tri := \{Tri_1, Tri_2, \ldots, Tri_n\}$. The remaining sorted triangle sequence is the resultant visibility sorting sequence.



Bold lines: Front polygon

Figure 4. A schema of procedure of visibility sorting.

3.2.3 Visibility Sorting of TPM: Visibility sorting of $Tpr := \{Tpr_1, Tpr_2, ..., Tpr_n\}$ from a 3-D viewpoint O can be executed in the following steps.

Step 1: Generate base triangles $Tri := \{Tri_1, Tri_2, ..., Tri_n\}$. Let the 2-D viewpoint of O be o.

Step 2: Execute visibility sorting for Tri, according to the procedure described in 3.2.2.

Step 3: Sort the prisms in the same order of sorted sequence of corresponding base triangles.

4. TRUE ORTHOIMAGE GENERATION BASED ON VISIBILITY SORTING

4.1 Generation of a True Orthoimage

Visibility sorting can be utilized in true orthoimage generation in the following sequences.

Step 1: Generate TPM and execute visibility sorting.

Step 2: Prepare output orthoimage buffer and occlusion buffer on the memory space and initialize them with value 0 (see Figure 5 (1)). Occlusion buffer should have the same width and height with the original image.

Step 3: Get the first prism from the visibility sorting sequence. Generate orthoimage for the top of the prism. In the occlusion buffer, record the area where the prism is projected in the original image, with non-zero pixel value (see Figure 5 (2)).

Step 4: Get the next prism and execute the same procedure in the Step 3. Before resampling of an orthoimage pixel from the original image pixel, check the corresponding occlusion buffer pixel if it is 0. If not, the orthoimage pixel should remain as the value 0. If the resampled pixel value is 0, change it to some proper low pixel value, such as 1, to distinguish it from pixels in occluded area. Repeat this step until all prisms are processed (see Figure 5 (3)-(6)).

In the resultant orthoimage, pixels of occluded area remain value 0.

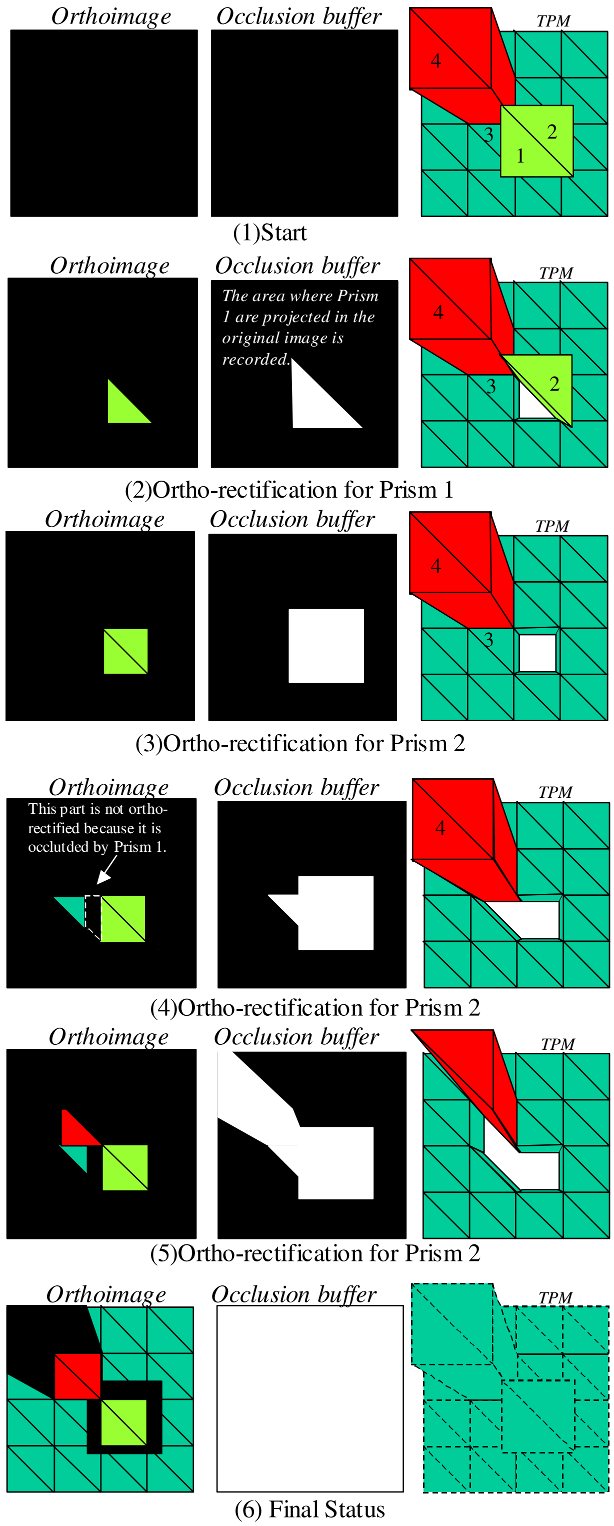


Figure 5. The schema of procedure of true orthoimage generation.

4.2 Generation of a Composite True Orthoimage

True orthoimages can be synthesized into one composite true orthoimage where occluded areas of one image are compensated with other true orthoimages. For each pixel coordinates (i, j), Composite true orthoimage $\overline{Iorth(i, j)}$ can be created from true orthoimages $Iorth_k(i, j)$ with the following equation:

$$\overline{Iorth(i, j)} = \frac{\sum_{k=1}^{nort} w(i, j, k) \times Iorth_k(i, j)}{\sum_{k=1}^{nort} w(i, j, k)}$$

where $w(i, j, k)$ is any proper weight function which satisfies:

$$Iorth_k(i, j) = 0 \Rightarrow w(i, j, k) = 0$$

5. TEST AND RESULTS

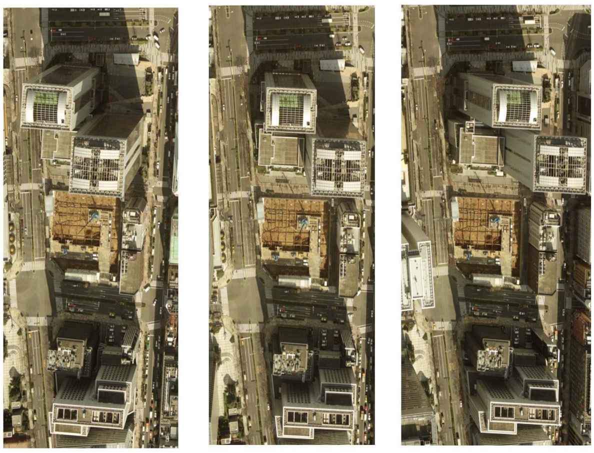
The algorithm has been tested with aerial images and digital map data shown in Table 1. Building polygons are extracted from LIDAR data automatically (Oda, K., et. al., 2004). The top three images in Figure 6 shows original images. True orthoimages are shown in Figure 6 (2), where occluded areas are filled with black colour. Figure 6 (3) shows the composite true orthoimage where occluded areas compensated with other true orthoimages. For this sample with 1589×2797 pixel^2, processing time of the true orthoimages is 1.25min per image with a Pentium 4 processor. The composite image is synthesized from 6 true orthoimages and its processing time is 0.66min.

Table 1. Data used in the test.

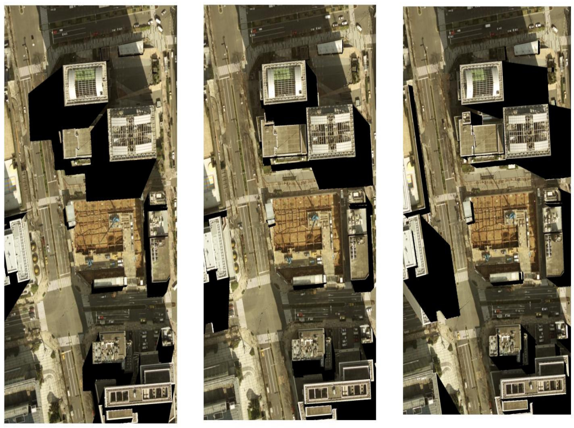
Items		Attributes
Target area		Shinjuku, Tokyo(1.2km×600m)
Aerial Images	Number/Courses	30 photos / 3 courses 1/8000 (80% overlap and 60% sidelap)
	Scale	1/8000
	Overlap/ Sidelap	80% / 60%
Digital map	Features on the ground	DEM created from LIDAR data
	Features above the ground.	Building polygons automatically extracted from LIDAR data and edited with a digital stereo plotter (Oda, K., et. al., 2004).

.**References**

Oda, K., Takano, T., Doihara, T., and Shibasaki, R., Automatic Building Extraction and 3-D City Modeling from LIDAR data Based on Hough Transromation, ISPRS2004, Commission III, PS WG III/8.



(1) Original images



(2) True orthoimages

(3) Composite true orthoimage
Figure 6. A sample of true orthoimage generation.