The Photogrammetric Record 22(118): 151–163 (June 2007)

MINIMISING ALGORITHM-INDUCED ARTEFACTS IN TRUE ORTHO-IMAGE GENERATION: A DIRECT METHOD IMPLEMENTED IN THE VECTOR DOMAIN

Yongwei Sheng (ysheng@geog.ucla.edu)
University of California, Los Angeles

Abstract

True ortho-images are generated from perspective images with both object and terrain displacement distortion corrected and with occlusions removed. However, widely used algorithms based on a Z-buffer have drawbacks and may result in artefacts in the generated ortho-images. This paper analyses the artefacts induced by Z-buffer algorithms in true ortho-image generation, and proposes a vector-based method to avoid algorithm-induced artefacts. The proposed method treats a perspective image pixel as a square patch instead of an image pixel point, and generates ortho-images in the vector domain. Under the vector implementation, occlusion detection is a simple process, and image resampling becomes a straightforward procedure. Most importantly, this method can minimise artefacts in the ortho-images so generated.

Keywords: image artefacts, occlusion detection, true orthophotos, vector-based resampling, *Z*-buffer

Introduction

Although aerial photographs carry a great deal of information, they are geometrically distorted and cannot be directly analysed with maps or managed in geographical information systems (GIS) due to the incompatible geometry. To remove the geometric distortion, they are transformed from the perspective geometry to the map-compatible orthographic projection to produce orthophotos through the process known as orthorectification. Orthophotos are image maps, maintaining the map geometry and containing the pictorial information of aerial photographs. Being produced by computer systems, digital orthophotos (or ortho-images) have advantages over their analogue counterparts, especially with respect to flexibility, possible derived products and integration with other data-sets (Baltsavias, 1996). Digital ortho-image production using computers was initiated in the late 1970s (Keating and Boston, 1979; Konecny, 1979). Being the core of the ortho-image generation process, rectification algorithms transform coordinates between the perspective image space and the ortho-image space based on the photogrammetric collinearity equations. Digital orthorectification transforms the perspective projection of an aerial photo or a satellite image into the orthographic map projection, and eliminates influences of terrain relief displacement as well as camera tilt. The production of digital ortho-images is a fairly simple process at the operational stage (Mehlbreuer et al., 1998).

Ortho-image generation becomes complicated when dealing with large scale photographs. Individual 3D objects such as buildings, bridges and trees become visible in large scale photographs or high-resolution images. If only terrain relief distortion is corrected in the orthorectification process, these high-rise objects remain distorted, leaning radially away from the principal point of the photographs, in the resulting ortho-images. This effect becomes more severe for taller objects closer to the edges of aerial photographs. Therefore, the distortion caused by object relief must be removed to produce large scale orthophotos of high quality.

The process of orthophoto production removing both terrain and object relief displacement distortion, which was initiated in the 1990s, is known as "true orthophoto" generation (Jensen, 1996; Amhar et al., 1998; Schickler and Thorpe, 1998; Rau et al., 2002; Sheng et al., 2003). True orthophoto generation usually involves multi-view images covering the same area. Since high-rise objects often cause occlusions in large scale photographs, the true orthophoto generation process takes pixel information from the non-occluded portion of the image viewed as vertically as possible (Sheng et al., 2003; Zhou et al., 2004). Therefore, occlusions need to be detected on individual images. A major technical difference between a true ortho-image generation algorithm and a conventional one is that the true ortho-image generation algorithm has the capability of detecting occlusions. *Z*-buffer techniques are widely used for occlusion detection in true ortho-image generation. However, *Z*-buffer algorithms may cause artefacts such as isolated points or lines in the ortho-images so generated (Rau et al., 2002; Sheng et al., 2003). These artefacts can be reduced by a subsequent filtering process (Amhar et al., 1998; Rau et al., 2002; Sheng et al., 2003), but this degrades ortho-image quality and cannot completely remove the artefacts. This paper analyses the sources of these artefacts and proposes a method implemented in the vector domain to avoid them.

Z-BUFFER TECHNIQUES AND ARTEFACT SOURCES IN ORTHO-IMAGERY

Ortho-image generation requires three sources of inputs: a perspective image, photo orientation parameters and a digital surface model (DSM) or a digital elevation model (DEM). Each input is critical to the quality of the resulting ortho-image. It is impossible to produce a good ortho-image using a perspective image of poor quality. In addition, inaccurate orientation parameters and poor DSMs can degrade ortho-image quality significantly (Wiesel, 1985; Höhle, 1996). Even though the input data is perfect, some algorithms may degrade the ortho-image generated by them and result in artefacts. Minimising algorithm-induced artefacts in true ortho-image production is the focus of this paper.

The occlusion problem needs to be addressed in true orthophoto generation, and occlusion-detection algorithms often induce artefacts in the resulting ortho-images. *Z*-buffer techniques in computer graphics for hidden surface removal (Pokorny and Gerald, 1989) are widely used for occlusion detection in true ortho-image generation (Amhar et al., 1998; Rau et al., 2002; Sheng et al., 2003). They are based on the fact that the front object occludes the objects behind. The camera-to-object depth information is recorded in a depth buffer (*Z*-buffer) to be used in visibility determination. Fig. 1 illustrates the principle of *Z*-buffer algorithms in true ortho-image generation. The *Z*-buffer array is defined in the perspective image space and initiated with a very large number; for every pixel in the perspective image it records the minimum distance between its corresponding surface point and the camera. A visibility buffer array is defined in the ortho-image space to record the visibility information for each pixel in the ortho-image. The algorithm starts from the ortho-image. For an ortho-image pixel *OP*, the 3D ground coordinates of its corresponding surface point *SP* are interpolated from the DSM, and its coordinates in the perspective image are computed using the photogrammetric collinearity equations and rounded to the nearest pixel *PP*. Pixel *OP* is considered the corresponding pixel in the ortho-image of the pixel *PP* in the perspective image. In the mean time, the distance between the camera lens centre *C* and *SP* is calculated and compared with

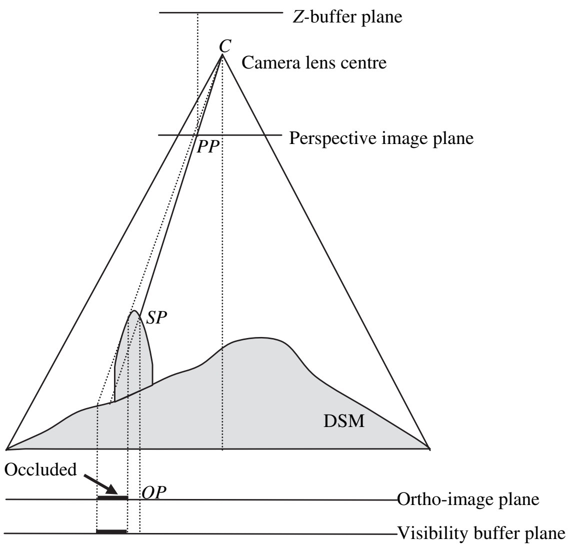

FIG. 1. Illustration of *Z*-buffer algorithms in true ortho-image generation.

the current distance value recorded in the *Z*-buffer array at pixel *PP*. If the calculated distance is greater, then the pixel *OP* is flagged as hidden in the visibility array; otherwise, the *Z*-buffer is updated with the new distance and the pixel in the visibility array previously recorded by the *Z*-buffer at pixel *PP* is flagged as hidden. After this process is repeated for each pixel in the ortho-image, the visibility buffer array records all the occluded areas. Pixel values in the perspective image are assigned only to the corresponding pixels in the ortho-image that are flagged as visible in the visibility array.

General *Z*-buffer algorithms in computer graphics are valuable for their simplicity and computational efficiency, but they are particularly prone to aliasing problems (Glassner, 1994). As with its ancestors in computer graphics, the *Z*-buffer algorithm in true ortho-image generation may induce artefacts in the resulting ortho-images. The *Z*-buffer array has to be defined in the perspective image space with the same dimensions and resolution as the perspective image, whereas the visibility array has to be aligned with the ortho-image. The collinearity equations, presenting a non-linear transformation, do not impose a mathematically strict one-to-one correspondence between the ortho-image coordinate system and the perspective image coordinate system. As a consequence, two problems are associated with the *Z*-buffer method. First, more than one ortho-image pixel may be projected to a single cell in the *Z*-buffer array, even though they may not occlude each other. This leads to false occlusions. Second, some cells in the *Z*-buffer array may not have any ortho-image pixel projected onto them, and this leads to gaps in the buffer array.

Both the false occlusion and the gap effects may result in artefacts such as isolated points or lines in the ortho-image. As illustrated in Fig. 2(a), the back-viewed slope *AB* is nine pixels long in the ortho-image, but appears only three pixels long in the perspective image. According to the *Z*-buffer algorithm, only one of the projected pixels in a *Z*-buffer cell can be visible. Thus, six of them (hatched cells) are flagged as occluded (false occlusions) in the visibility array, but actually they all should be visible. As a consequence, the six pixels leave blanks in the ortho-image. On the other hand, six ortho-image pixels of

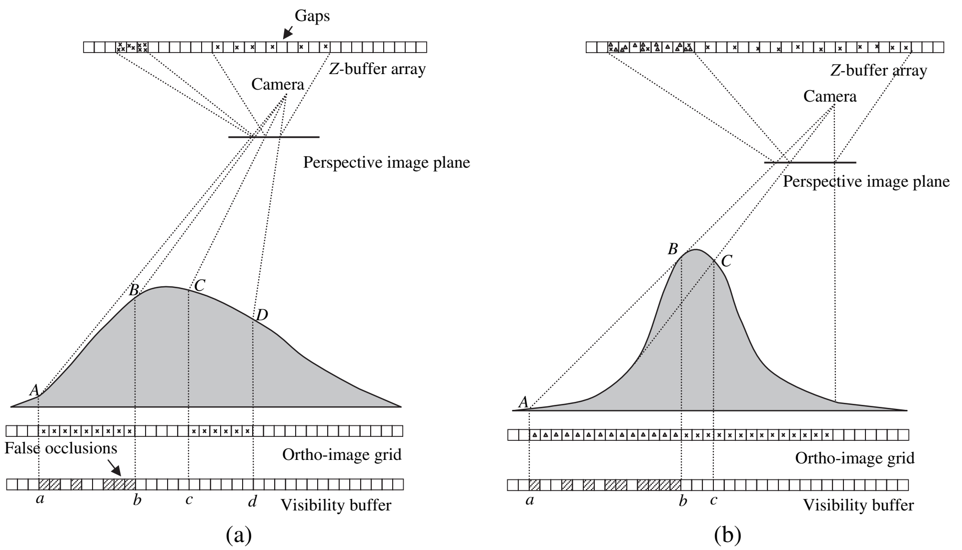

FIG. 2. Artefacts induced by *Z*-buffer algorithms: (a) false occlusion and gap effects; (b) gap effect interfering with occlusion detection.

the fore-viewed slope *CD* are projected to 11 pixels in the perspective image, leaving five empty cells (gaps) in the *Z*-buffer array. In cases where no occlusion occurs (Fig. 2(a)), the gaps in the *Z*-buffer array do not produce artefacts in the ortho-image at all since these gaps do not influence the ortho-image pixels. However, this is not true when occlusions are present. Occluded ortho-image pixels may be projected to these gaps and detected as visible. This situation is illustrated in Fig. 2(b). Since the slope *AB* is occluded by the slope *BC*, the corresponding ortho-image pixels within *ab* should be identified as occluded. The 14 occluded ortho-image pixels (represented by triangles) within *ab* and the three visible pixels (represented by crosses) within *bc* are projected to eight cells in the *Z*-buffer array. As a result, five cells in the buffer array do not have any pixels projected from the *bc* segment, but they are occupied by occluded pixels from the *ab* segment. As a result, the *Z*-buffer algorithm flags five of the occluded pixels as visible, though they are in fact occluded by the *BC* slope. This leads to artefacts in the occluded areas.

PROPOSED DIRECT METHOD IMPLEMENTED IN VECTOR DOMAIN

Based on the direction of the transformation, orthorectification algorithms are categorised into two groups of methods, direct and indirect (Chen and Lee, 1993; Mikhail et al., 2001; Sheng et al., 2003). Direct methods start from the perspective image space, while indirect ones start from the ortho-image. Direct methods iteratively calculate the ground coordinates of a perspective image pixel, and involve an image interpolation process that is not straightforward. As a result, indirect methods are advantageous over direct methods in conventional ortho-image generation (Mayr and Heipke, 1988; Chen and Lee, 1993); direct methods are in general used in ortho-image generation only when rigorous equations of coordinate transformation are not available, for example, for images acquired by scanning systems and satellite systems (O'Neill and Dowman, 1988; Chen and Lee, 1993; Li, 2001).

Though indirect methods incorporating *Z*-buffer techniques are widely used in true ortho-image generation, direct methods have their own strengths. The latter, also known as top-down methods, start with a pixel in the perspective image space, convert its image coordinates to ground coordinates by intersecting the viewing ray (that is, the projection ray) with the DSM surface, calculate its location in the ortho-image, and assign its image value to the ortho-image pixel found. In such a way, occluded ortho-image pixels are not assigned any values. Therefore, no explicit procedure is needed for occlusion detection.

Direct methods are conventionally implemented in the raster domain, from image (the perspective image) to image (the ortho-image). These methods, however, are also prone to artefacts, especially when fewer perspective image pixel values are assigned to ortho-image pixels. Some pixels in the ortho-image, though not occluded, may not be assigned any values. This leads to gaps in the ortho-images so produced. Though these gaps can be filled with neighbouring pixel values using interpolation methods, this will degrade the quality of the resulting ortho-image.

Implementation in the raster domain is a major cause of these algorithm-induced artefacts for both direct methods and indirect methods. The coordinate conversion is performed from image to image through a non-linear transformation, and a rounding process has to be introduced to convert coordinates to image row and column numbers, which are integers. These artefacts are therefore unavoidable in the raster implementation. Therefore, this paper proposes a direct method implemented in the vector domain to avoid these artefacts.

In the proposed vector-based direct method, a pixel in the perspective image is considered as a square patch instead of an image pixel point, and the pictorial value of the patch takes the perspective pixel value. The ground coordinates of each corner of a pixel (a square patch) in the perspective image are determined by a ground coordinate determination algorithm. Sheng (2004) evaluated three algorithms of this kind, and found that the iterative photogrammetric method is prone to the occlusion and divergence problems; the non-iterative ray-tracing method is computationally intensive; and the iterative ray-tracing method can offer a compromise between efficiency and effectiveness. Thus, the iterative ray-tracing algorithm was used in this paper to compute ground coordinates for perspective image pixels. Each pixel patch in the perspective image is projected to be a quadrilateral in the ground coordinate system. The horizontal projection of all the quadrilaterals from the perspective image forms a complete coverage over the ortho-image grids without overlaps or gaps (Fig. 3(a)). This is the key reason why the vector implementation is advantageous over its raster counterpart, and explains how it avoids artefacts.

These projected polygons completely cover the ortho-image plane including those occluded areas. Thus occlusions need to be detected. Occlusion detection is a simple process under the vector implementation. Occlusions occur only when a quadrilateral is back-viewed and overrides a high object and a low one. As illustrated in Fig. 3(b), a perspective pixel is projected to the quadrilateral *ABCD* on the ground, with its *AB* part on a higher object while the *CD* part lies on a lower object. The horizontal projection of *ABCD* on the ortho-image plane is *abCD*. Part of the *abCD* area in the ortho-image may be occluded. *AB* is projected to *EF* on the ortho-image plane through the view rays, and *abEF* (the cross-hatched polygon in Fig. 3(b)) represents the region in occlusion. Then the ortho-image pixels within *abEF* are detected as occluded. The hatched polygons in Fig. 3(a) illustrate the occluded areas.

There is no simple one-to-one mapping between the ortho-image pixels and the projected polygon patches. One ortho-image pixel may be covered by more than one patch, and one polygon patch may be occupied by more than one ortho-image pixel. Therefore, a resampling process is involved. The value of an ortho-image pixel is assigned according to its touching

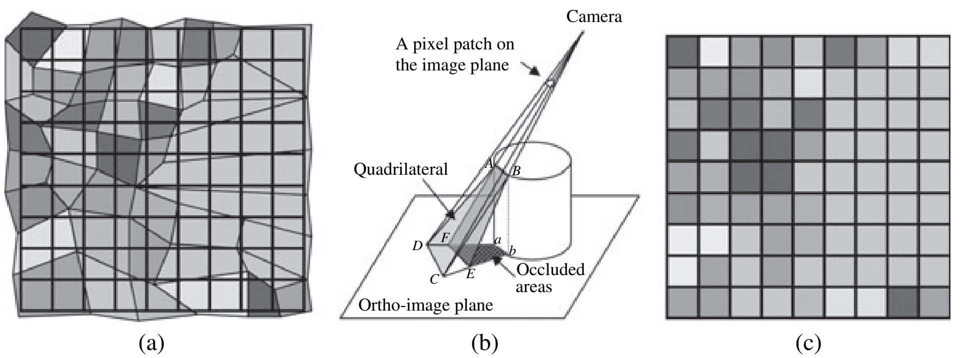

FIG. 3. Illustration of vector-based direct method: (a) projected polygons forming a complete coverage on an ortho-image; (b) occlusion detection in vector implementation; (c) ortho-image produced using centre-priority resampling scheme.

polygon patches through the resampling process. Under the vector implementation, there are three straightforward resampling methods: the centre-priority scheme, the dominant area scheme and the proportional area scheme. The centre-priority method is desirable for its simplicity. An ortho-image pixel takes its value from the projected perspective pixel patch in which its centre falls. Applying the centre-priority scheme to the polygon coverage in Fig. 3(a), an ortho-image can be produced in Fig. 3(c). In the two area-based resampling schemes, an ortho-image pixel is also treated as a square patch, and the area of each small polygon resulting from intersecting the ortho-image grid patch with the projected polygons needs to be calculated. In the dominant area scheme, the pixel value of the ortho-image takes the value of the perspective pixel patch with the greatest area contribution. Under the proportional area scheme, the pixel value is determined by the values of the perspective pixel patches partially covering this ortho-image pixel patch, and their contribution is proportional to their overlap area with the ortho-image pixel patch.

EXPERIMENTS

Study Site and Data Collection

The study site is a redwood stand approximately 166 m long and 122 m wide, containing about 60 redwood trees, several oak trees and parts of two large buildings. The topographic height range at this site is about 12 m, while the heights above ground level of the buildings and trees are up to 45 m. A 1:2400 scale vertical aerial photograph of the site taken with a camera of 152·8 mm focal length was scanned at a resolution of 250 dots per inch (dpi), making the pixel resolution approximately 0·24 m on the ground. The camera exposure station location and camera attitude of the photo were solved through photogrammetric orientation procedures. The study site is covered by the left portion (Fig. 4(a)) of the photo, on which trees and buildings are the major objects. The dominant object relief distortion and severe occlusions caused by both human structures and natural objects make this image ideal for testing true ortho-image generation algorithms. The canopy surface model of the stand was reconstructed from a model-based stereomatching scheme (Sheng, 2000). The building surface model was established interactively and was combined with the canopy surface model to produce a DSM of 0·24 m resolution for the study area (Fig. 4(b)).

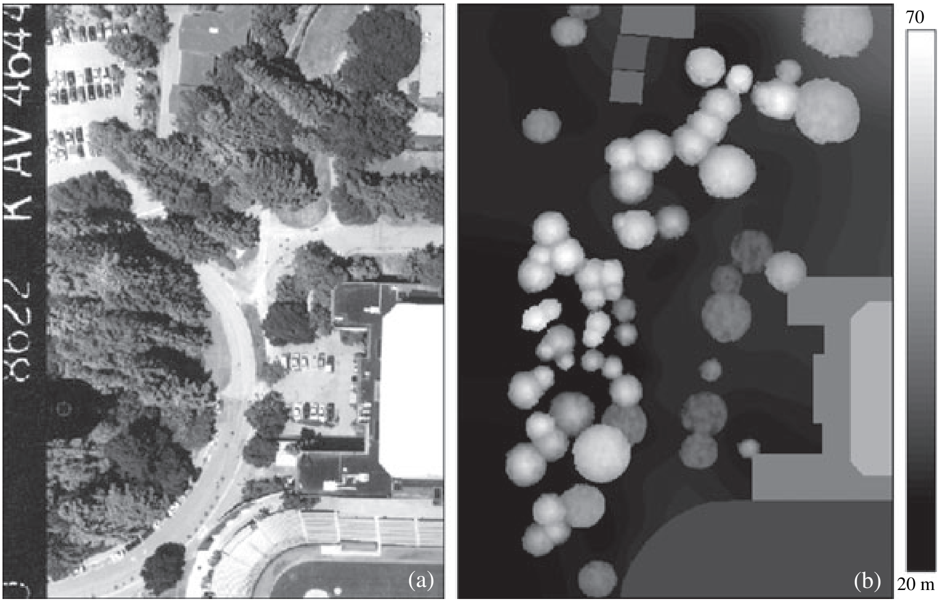

FIG. 4. Perspective image and surface model of study site: (a) perspective image; (b) digital surface model.

Analysis of Results

For a comparison, the true ortho-image generation algorithms discussed above were applied to the data-set to produce 0·24 m resolution ortho-images. Fig. 5(a) shows the ortho-image generated by the indirect method using a *Z*-buffer algorithm, and Fig. 5(b) magnifies the portion within the white box in Fig. 5(a). Artefacts are severe in the generated ortho-image. The striking black lines are the results of the false occlusion effect in the *Z*-buffer array. More disappointingly, the occlusion problem is not properly addressed. The areas to the left of many trees should actually be occluded by these trees, but the *Z*-buffer algorithm failed to detect them. This is because the gap effect in the *Z*-buffer array interferes with occlusion detection as illustrated in Fig. 2(b). It is obvious that the generated ortho-image is of poor quality.

The gap effect in the *Z*-buffer array can be considerably relieved by increasing the density of DSM sampling during the projection process. Applying the *Z*-buffer method to the DSM with three times greater sampling density, the resulting ortho-image and the zoom-in portion are shown in Figs. 6(a) and (b). Compared with the ortho-image produced at the original DSM sampling density, the results are significantly improved. Occluded areas were much better detected (though still not very well) and artefacts were considerably reduced. It is noticeable that the black lines do not change much. This suggests that denser DSM sampling can considerably reduce the gap effect in the *Z*-buffer array, but cannot relieve the false occlusion effect.

The raster-based direct method can produce even better results. Figs. 7(a) and (b) show the resulting ortho-image and close-up. Though no explicit occlusion detection algorithm was used, the occlusion problem was well addressed, with only a few erroneous pixels evident, even better than the *Z*-buffer-based algorithm with denser DSM sampling. The black lines are almost identical to those in Figs. 5 and 6. Leaving gaps or lines in the resulting ortho-images is a common problem of these raster-based methods.

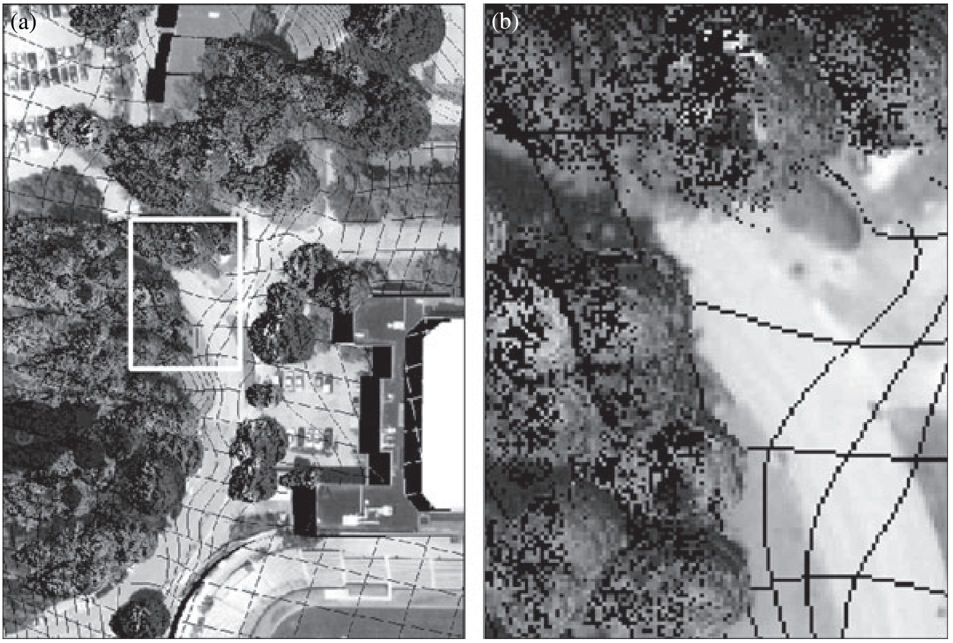

FIG. 5. Ortho-image generated by *Z*-buffer algorithm: (a) ortho-image; (b) close-up view of white box in (a).

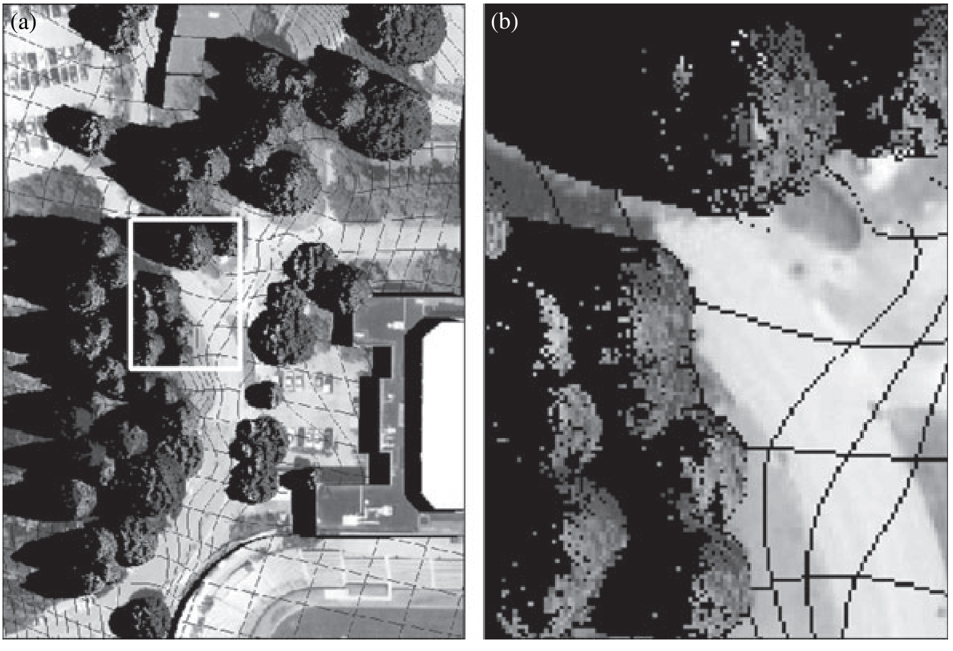

FIG. 6. Ortho-image generated by *Z*-buffer algorithm with increased-density DSM sampling: (a) ortho-image; (b) close-up.

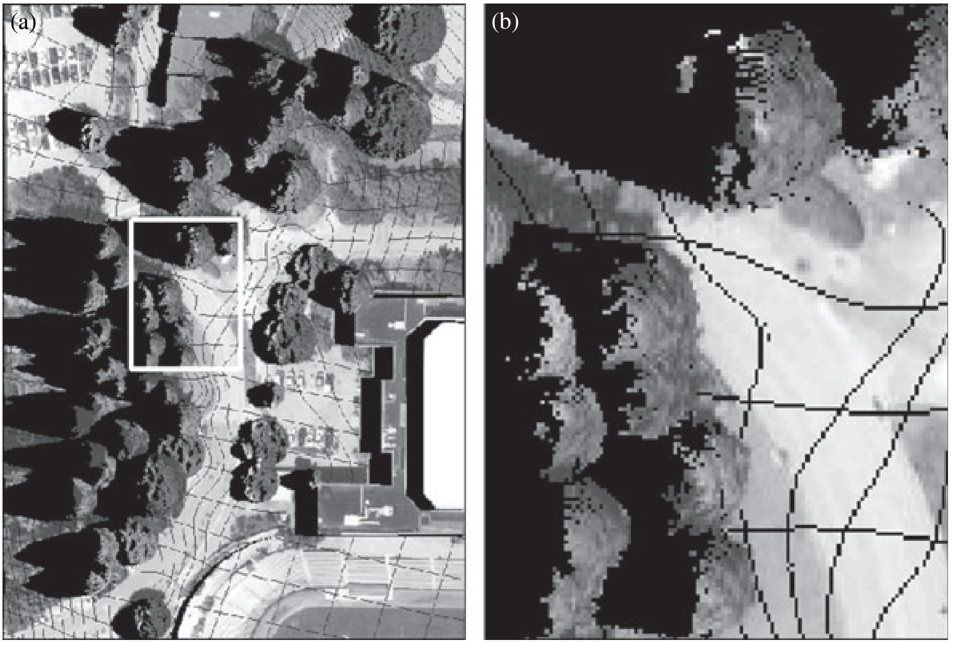

FIG. 7. Ortho-image generated by raster-based direct algorithm: (a) ortho-image; (b) close-up.

The proposed vector-based direct method can adequately address the artefact problem. Using the centre-priority resampling scheme, the proposed algorithm produced the ortho-image shown in Figs. 8(a) and (b) (close-up). The produced ortho-image is of high quality, without black lines or other remarkable artefacts. This demonstrates the effectiveness of the proposed method. Figs. 8(c) and (d) show the close-up portion of the ortho-images generated with the dominant area and proportional area resampling schemes, respectively. The ortho-images produced with the centre-priority and dominant area schemes are similar to each other, while that produced with the proportional area scheme tends to be smoother since an averaging process is involved.

Computational Efficiency Analysis

In addition to the quality of the ortho-images produced, the computational efficiency of the algorithms is also analysed. The indirect method based on the *Z*-buffer is the most economic because it inherits its efficiency from both indirect methods and *Z*-buffer algorithms. Since computing ground coordinates for perspective image pixels is a slow process, the direct methods are less efficient compared to the indirect methods based on the *Z*-buffer. Most costs of the direct methods (99% for the raster implementation and 88% for the vector implementation) are spent on ground coordinate calculation. The vector-based direct method can produce artefact-free ortho-images with about 11% additional computational costs compared to the conventional raster implementation. Among these three resampling schemes under the vector implementation, the centre-priority scheme is the most efficient. The two area-based resampling methods are most computationally intensive because intersection needs to be

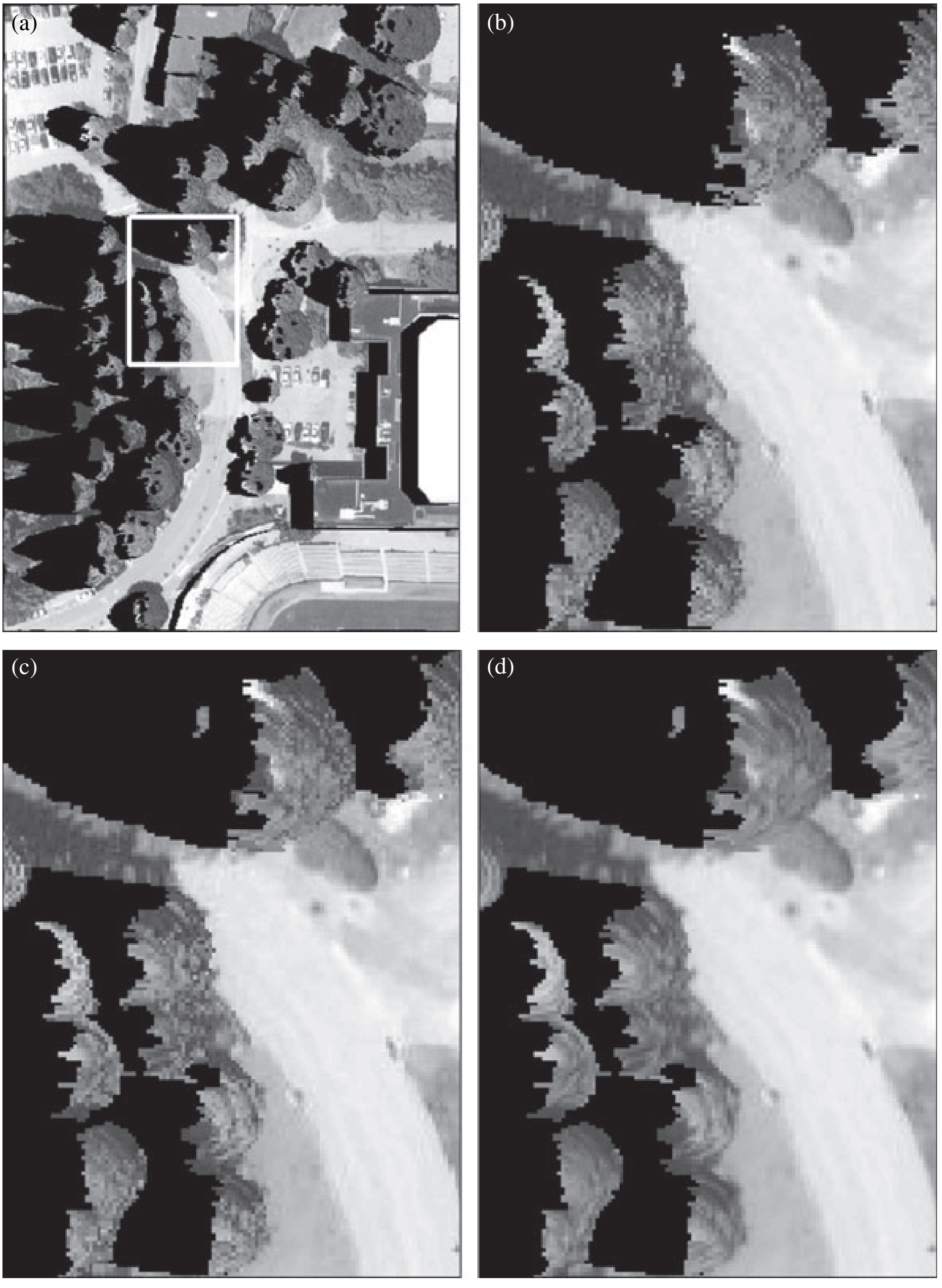

FIG. 8. Ortho-image generated by vector-based direct algorithm:
(a) ortho-image produced using centre-priority resampling scheme; (b) close-up view of white box in (a);
(c) close-up view of ortho-image produced using dominant area resampling scheme;
(d) close-up view of ortho-image produced using proportional area resampling scheme.

computed between the projected polygon patches and ortho-image grids, and the area of each resultant small polygon also needs to be calculated.

Discussions and Conclusions

The occlusion problem in true ortho-image rectification has to be addressed, and algorithms for this often induce artefacts in the resulting ortho-images. This paper discusses algorithm-induced artefacts in true ortho-image generation, and analyses the widely used indirect method based on a *Z*-buffer. This method works best when the ortho-image pixels and the perspective image pixels are in a one-to-one correspondence, which is an ideal case. Otherwise, the false occlusion and the gap effects of the *Z*-buffer algorithm produce artefacts in the ortho-images. The gap effect can be considerably reduced by increasing the density of DSM sampling, but cannot be completely removed. It is impossible to reduce the false occlusion effect since more than one pixel, though not occluded, may be projected to a same cell in the *Z*-buffer array.

Direct ortho-image generation methods are traditionally implemented in the raster domain. Though indirect methods are considered advantageous over direct ones in producing conventional ortho-images, direct methods have their strengths in true ortho-image generation due to their suitability in addressing the occlusion problem. Though no explicit occlusion detection is imposed, the conventional raster implementation can better handle the occlusion problem and induce fewer artefacts than the *Z*-buffer algorithm even with increased-density DSM sampling. However, the raster-based direct method may still result in gaps in the ortho-images where no perspective pixels are projected, and these gaps are unavoidable in a raster-based implementation.

This paper proposes a vector-based direct method to minimise algorithm-induced artefacts. This method treats a perspective image pixel as a square patch instead of an image pixel point, and generates ortho-images in the vector domain. The proposed vector-based method is simple and effective. Occlusion detection is a simple process in the vector implementation, and this method can avoid artefacts in the resulting ortho-image. The proposed vector-based direct algorithm demonstrated its great ability to produce artefact-free ortho-images. As a direct method, it is applicable to images acquired by various remote sensing systems, not limited to photogrammetric cameras. Though it is computationally intensive compared to the *Z*-buffer-based algorithm, the proposed vector-based direct method with the centre-priority resampling scheme is affordable, costing about 11% additional computation compared to the raster-based direct method.

References

Amhar, F., Jansa, J. and Ries, C., 1998. The generation of true orthophotos using a 3D building model in conjunction with a conventional DTM. *International Archives of Photogrammetry and Remote Sensing*, 32(4): 16–22.

Baltsavias, E. P., 1996. Digital ortho-images—a powerful tool for the extraction of spatial- and geo-information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 51(2): 63–77.

Chen, L.-C. and Lee, L.-H., 1993. Rigorous generation of digital orthophotos from SPOT images. *Photogrammetric Engineering and Remote Sensing*, 59(5): 655–661.

Glassner, A. S., 1994. *3D Computer Graphics: A User's Guide for Artists and Designers*. Second edition. Design Books, New York. 213 pages.

Höhle, J., 1996. Experiences with the production of digital orthophotos. *Photogrammetric Engineering and Remote Sensing*, 62(10): 1189–1194.

JENSEN, J. R., 1996. Issues involving the creation of digital elevation models and terrain corrected orthoimagery using soft-copy photogrammetry. *Digital Photogrammetry: An Addendum to the Manual of Photogrammetry* (Ed. C. W. Greve). American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland. 247 pages: 167–179.

KEATING, T. J. and BOSTON, D. R., 1979. Digital orthophoto production using scanning microdensitometers. *Photogrammetric Engineering and Remote Sensing*, 45(6): 735–740.

KONECNY, G., 1979. Methods and possibilities for digital differential rectification. *Photogrammetric Engineering and Remote Sensing*, 45(6): 727–734.

LI, Z., 2001. Orthoimage generation and measurement from single images. *Geographical Data Acquisition* (Eds. Y.-Q. Chen and Y.-C. Lee). Springer, Vienna. 265 pages: 173–192.

MAYR, W. and HEIPKE, C., 1988. A contribution to digital orthophoto generation. *International Archives of Photogrammetry and Remote Sensing*, 27(B11): IV430–IV439.

MEHLBREUER, A., SCHROTH, R., WEIDNER, U. and AL ZAFFIN, M., 1998. Digital colour orthophotos for boomtown Dubai. *International Archives of Photogrammetry and Remote Sensing*, 32(4): 367–374.

MIKHAIL, E. M., BETHEL, J. S. and MCGLONE, J. C., 2001. *Introduction to Modern Photogrammetry.* John Wiley, New York. 479 pages.

O'NEILL, M. A. and DOWMAN, I. J., 1988. The generation of epipolar synthetic stereo mates for SPOT images using a DEM. *International Archives of Photogrammetry and Remote Sensing*, 27(B3): 587–598.

POKORNY, C. K. and GERALD, C. F., 1989. *Computer Graphics: The Principles behind the Art and Science*. Franklin, Beedle & Associates, Irvine, California. 738 pages.

RAU, J.-Y., CHEN, N.-Y. and CHEN, L.-C., 2002. True orthophoto generation of built-up areas using multi-view images. *Photogrammetric Engineering and Remote Sensing*, 68(6): 581–588.

SCHICKLER, W. and THORPE, A., 1998. Operational procedure for automatic true orthophoto generation. *International Archives of Photogrammetry and Remote Sensing*, 32(4): 527–532.

SHENG, Y., 2000. *Model-based conifer crown surface reconstruction from multi-ocular high-resolution aerial imagery.* Ph.D. dissertation, University of California at Berkeley, California. 168 pages.

SHENG, Y., 2004. Comparative evaluation of iterative and non-iterative methods to ground coordinate determination from single aerial images. *Computers & Geosciences*, 30(3): 267–279.

SHENG, Y., GONG, P. and BIGING, G. S., 2003. True orthoimage production for forested areas from large scale aerial photographs. *Photogrammetric Engineering and Remote Sensing*, 69(3): 259–266.

WIESEL, J., 1985. Digital image processing for orthophoto generation. *Photogrammetria*, 40(2): 69–76.

ZHOU, G., SCHICKLER, W., THORPE, A., SONG, P., CHEN, W. and SONG, C., 2004. True orthoimage generation in urban areas with very tall buildings. *International Journal of Remote Sensing*, 25(22): 5163–5180.

Résumé

Pour réaliser des ortho-images rigoureuses à partir d'images perspectives, il convient de corriger à la fois les objets et le terrain des déformations et des défauts de mise en place et d'éliminer les parties cachées. Toutefois les algorithmes généralement utilisés qui s'appuient sur un Z-buffer présentent divers inconvénients qui peuvent occasionner des artéfacts dans les ortho-images correspondantes. On étudie dans cet article les artéfacts que de tels algorithmes peuvent introduire dans la confection d'ortho-images rigoureuses et on propose une méthode orientée vecteur capable d'éviter ces défauts. Dans cette méthode, on considère que chaque pixel de l'image perspective n'est pas un élément ponctuel mais un élément surfacique carré et on génère les ortho-images en mode vectoriel. En effectuant tous les développements dans ce mode vectoriel la détection des parties cachées est simplifiée et le ré-échantillonnage de l'image devient un processus facile. Mais le plus important reste que cette méthode peut minimiser les artéfacts dans les ortho-images ainsi confectionnées.

Zusammenfassung

Wahre Orthobilder werden aus perspektiven Aufnahmen durch Korrektur der Reliefversetzung von Gelände und Objekten und Beseitigung von Verdeckungen *erzeugt. Gängige Algorithmen, die sich auf die Z-Puffer Methode stützen, haben Nachteile und können Artefakte in den generierten Orthobildern hervorrufen. Diese Artefakte werden zunächst analysiert und es wird dann eine vektorbasierte Methode vorgestellt, die solche Artefakte vermeidet. Diese Methode behandelt jedes Pixel in der perspektiven Aufnahme als quadratische Fläche und nicht als Bildpunkt und generiert Orthobilder im Vektorraum. Damit ist die Erkennung von Verdeckungen vereinfacht und die Umbildung wird ebenfalls zu einer einfachen Prozedur. Wichtig hierbei ist jedoch die Minimierung von Artefakten in den generierten wahren Orthobildern.*

Resumen

Las ortoimágenes verdaderas se generan a partir de imágenes perspectivas corrigiendo la distorsión del desplazamiento de los objetos y del terreno, y eliminando las oclusiones. Sin embargo, los algoritmos comunes basados en un Z-buffer tienen limitaciones que pueden originar artefactos en las ortoimágenes generadas. Este artículo analiza los artefactos inducidos por los algoritmos Z-buffer en la generación de ortoimágenes verdaderas, y propone un método basado en vectores para evitar artefactos inducidos por el algoritmo. El método propuesto trata al píxel de la imagen perspectiva como si fuera un cuadrado en lugar de como un punto de imagen, y genera ortoimágenes en el dominio vectorial. Utilizando vectores, la detección de oclusiones y el remuestreo de la imagen se convierten en procesos sencillos, y lo que es más importante, se minimizan los artefactos en las ortoimágenes generadas.

