# DEM GENERATION FROM VERY HIGH RESOLUTION STEREO SATELLITE DATA IN URBAN AREAS USING DYNAMIC PROGRAMMING

Thomas Krauß[1], Peter Reinartz[1], Manfred Lehner[1], Manfred Schroeder[1], Uwe Stilla[2]
[1]German Aerospace Center (DLR), Remote Sensing Technology Institute
PO Box 1116, 82230 Wessling, Germany
thomas.krauss@dlr.de
[2]Photogrammetry and Remote Sensing, Technische Universitaet Muenchen
Arcisstrasse 21, 80333 Muenchen, Germany
stilla@bv.tum.de

**KEYWORDS:** VHR data, Stereo Data, DEM generation, dynamic programming, dynamic time warping

**ABSTRACT:**

This article shows first results of implementing a method for creating DEMs from high resolution satellite imagery based on dynamic programming. The herein described DTW algorithm maps epipolar stereo image pairs line by line on top of each other using a method similar to dynamic time warping which is a common approach in speech recognition. The DTW algorithm is described and applied to several test images. The resulting DEMs and pros and cons of this method are shown and discussed.

## 1 INTRODUCTION

Images delivered from actual very high resolution (VHR) satellites like Ikonos or QuickBird provide resolutions of about one meter. Object details at a size of approximately one meter can not clearly be distinguished and limit the accuracy of the DEM evaluation process. Furthermore urban scenes show steep walls and narrow streets demanding new approaches in creating sufficent DEMs automatically. As shown in a previous paper (Krauß et al. (2005)) „classical“ DEM generation algorithms based on pattern matching like Lehner and Gill (1992) fail in urban areas due to steep elevation changes and many obscured regions. Such algorithms work better with relatively smooth height changes without occlusions which can be found in satellite imagery of lower resolution. On the other hand an object oriented approach like Schenk (2004) fails due to missing larger homogeneous regions. Such methods work much better on images with higher resolution like aerial images.

So the intention of this investigation is to find an algorithm reconstructing steep objects in a scene with feature variations in the order of magnitude of the given image resolution. Starting with the „column“ algorithm from the previous paper for the idea of mapping epipolar lines in each of the two images of a stereo pair an improved method has to be found.

By visual interpretation of a stereo image pair, objects like houses, trees and cars can be interpreted easily and steep features down to the size of one pixel can be detected by eye without problems. Also the human recognition masks out inconsistent areas automatically. So hidden features in one and the other image vanish when looking at the stereoscopic image and the brain recognizes a fairly good ortho image with nearly perfect height information.

But how is this done by the human recognition process? Observing in detail what is happening if one looks on an extreme stereo red/green stereo image with red/green glasses shows that in the first moment the eye is irritated and attempts to get the images one on to the other. This feels as if the eye takes the two images as rubber sheets and attempts to flex the images in similar ways since they match. If this first adaption is done more and more details in height differences can be detected.

Based on this observation an algorithm was searched which maps corresponding lines of the image pair one to the other. The restriction to only lines can be introduced if we stick to epipolar geometry of the images.

A possible solution was found in speech recognition. Since recorded samples of words rarely fit to actual spoken words a system has to be used matching the actual voice with learned words. To achieve this all samples are fragmented in overlapping short parts of which a spectral characteristic is memorized. The same procedure is applied to the voice input. Since speach varies in duration such sequences seldom fit together. But a simple linear stretching in time is not applicable since words are not uniformly stretched. Vowels are stretched more wheras consonants mostly keep the same length. Based on these facts an algorithm referred to as „dynamic time warping“ was developed which is based on dynamic programming (Schiele (2005), Culjat (1999)).

This algorithm was taken as base and adapted to stereo images as shown consecutively.

## 2 DESCRIPTION OF THE ALGORITHM

The algorithm is based on epipolar images. Such images are for example processed Ikonos stereo pairs or the reference images from Scharstein and Szeliski (2002). The images are assumed to be available as a stereo pair with parallaxes in image line direction (left to right, horizontal epipolar lines). Image rows (lines) are horizontally (left to right), image columns vertically (top to bottom). Examples of such image pairs are shown in figures 4 and 9.

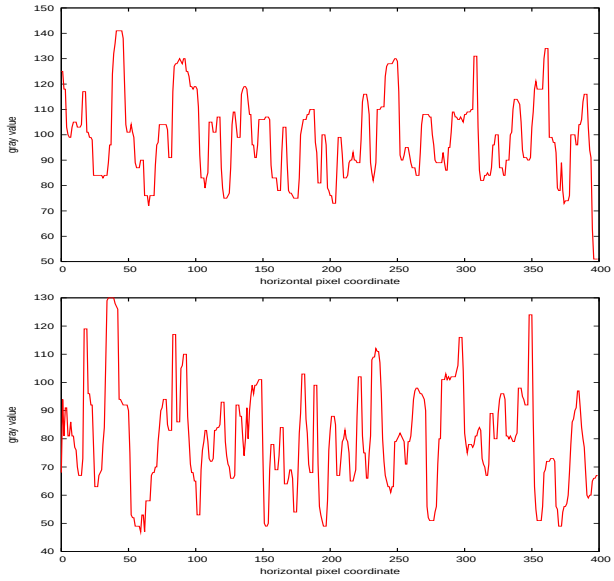


Figure 1. Two corresponding lines of the stereo pair shown in 4 as sequences of grey values.

Creating a digital elevation model (DEM) from two images is

based on finding corresponding image parts in each of the two images and calculating a local height of an image part out of the relative displacement of the locations of this part in the two images (parallaxes).

Because of the assumption of epipolar images there exist only horizontal parallaxes and no vertical shifts between the two images. This allows reducing the problem „find correlations of image parts between two images“ to only „find correlations in corresponding two lines of the images“.

Each line of an image is represented as a sequence of gray values as shown in figure 1.

Such sequences have to be mapped on each other stretching and compressing parts to achieve an optimal local fit. This problem is very common in signal processing and well known in speech recognition as „dynamic time warping“ (DTW, Schiele (2005), Culjat (1999)). Also some authors in photogrammetry suggest similar methods for feature based image matching (Baltsavias (1997)). In this paper the DTW method used in speech recognition for comparing recorded speech sequences with a library of words is modified for comparing directly two sequences of gray values instead of derived features.

The DTW method used in speech recognition calculates spectral characteristics for short overlapping parts of the audio samples, calculate distances between each of these parts and uses dynamic programming to receive a so called „minimal total distance“ for the given sample with the compared sample of the dictionary. Only this minimal total distance is used further for determining the most probable sequence of words.

This distance is a measure for all needed shifting, stretching and squeezing operations for one sequence to fit onto the other. For speech recognition it's sufficient finding the dictionary sample with the smallest minimal total distance to the given voice input. But beneath this minimal total distance a so called „minimal path“ can be defined. This path connects the endpoints of the compared sequences in a matter describing what parts of one sequence has to be shifted, streched or squeezed to fit on the other. From this minimal path the parallaxes we are searching for can be extracted.

## 3 IMPLEMENTATION OF THE ALGORITHM

For the implementation of the algorithm we first have to define a „distance“. In the case of a direct comparision of the sequences of gray values of the two epipolar images this can be as simple as the gray value distance rather than first calculating parameters and defining some distances upon these.

For clarification let's take as an example two sequences of gray values $I$ and $I'$ as:

$$I = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad I' = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}$$

The matrix $M_{i,j}$ will then be calculated as

$$M_{i,j} = \left| I_i - I'_j \right|$$

thus

$$M = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 1 \\ 2 & 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 1 \end{pmatrix}$$

In the next step the rows and columns are cumulated to a matrix $D$ filling the first line and column according to

$$D_{1,j} = \sum_{k=1}^{j} M_{1,k}, \qquad D_{i,1} = \sum_{k=1}^{i} M_{k,1}$$

and the rest of the matrix $D_{i,j}$ $(i, j > 1)$ according to

$$D_{i,j} = M_{i,j} + \min\left\{ D_{i-1,j}, D_{i,j-1}, D_{i-1,j-i} \right\}$$

yielding $D$ in the example from above:

$$D = \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 1 & 2 & 1 & 3 & 4 \\ 3 & 2 & 3 & 1 & 2 \\ 4 & 2 & 3 & 2 & 1 \\ 4 & 3 & 2 & 4 & 2 \end{pmatrix}$$

In this Matrix $D$ the overall distance is defined as the rightmost bottom element – in the example 2. Starting from this element going back always using the smalles possible next neighbour to the top, left or top-left gives the following minimal path (marked in bold):

$$D = \begin{pmatrix} \mathbf{1} & \mathbf{1} & 2 & 3 & 3 \\ 1 & 2 & \mathbf{1} & 3 & 4 \\ 3 & 2 & 3 & \mathbf{1} & 2 \\ 4 & 2 & 3 & 2 & \mathbf{1} \\ 4 & 3 & 2 & 4 & \mathbf{2} \end{pmatrix}$$

Taking the gray value profiles from figure 1, calculating the matrix $D_{i,j}$ and marking this minimal path creates the path shown in figure 3 ($i, j$ run in this figure from the bottom left to the upper right corner).

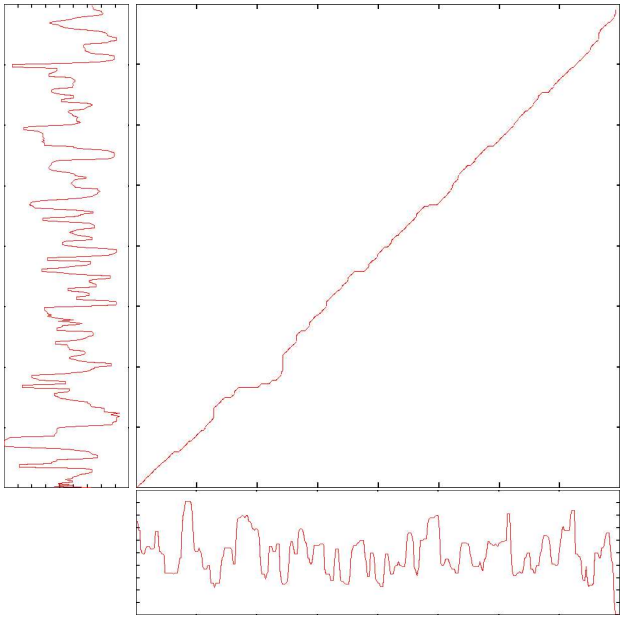

Figure 2. Minimal path created by the DTW algorithm with the grey value profiles from fig. 1 as inputs (shown on the left and bottom for better correlation)

Picking a small area of this result and showing the extracted correlations between the input gray value profiles yields fig. 3. As can be seen well areas with different widths in the profiles are correctly mapped to each other.

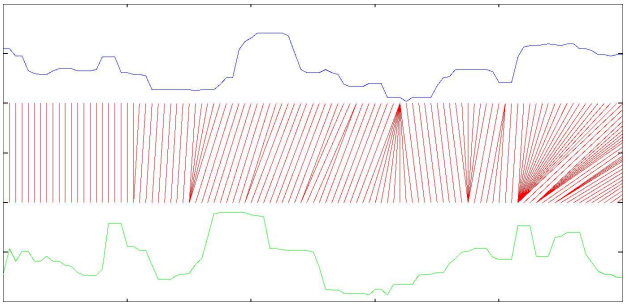

Figure 3. Found correlations between the two input gray value profiles (top and bottom of graphic)

A perfectly diagonal line in fig. 2 represents no displacement between the two profiles. All deviations from this line indicate the searched parallaxes. If this minimal path is represented by pairs of coordinates $\mu_k = (i,j)_k$ the horizontal position $(i+j)/2$ corresponds with the parallax $i-j$ if the stereo images are taken with symmetric inclination angles.

Using these parallaxes found for each position in the horizontal lines an ortho image can be calculated by averaging the corresponding gray values of the two input gray value profiles at the parallax positions.

Using small window areas of diameter $w$ around every line position for calculating the distance $M_{i,j}$ gives better results as shown later. The distance $M_{i,j}$ of line $y$ with images $I_{x,y}$ and $I'_{x,y}$ and window size $w$ is then calculated as

$$M_{i,j} = \sum_{\lambda=-\lfloor w/2 \rfloor}^{w-\lfloor w/2 \rfloor -1} \sum_{\mu=-\lfloor w/2 \rfloor}^{w-\lfloor w/2 \rfloor -1} \left( I_{i+\lambda,y+\mu} - I'_{j+\lambda,y+\mu} \right)$$

In words: over a window of size $w \times w$ pixels all gray value distances between the corresponding pixels in the two images are summarized ($\lfloor x \rfloor$ is the largest integer smaller than $x$).

For acceleration of processing time a maximum correlation window size can be choosen. Outside of this distance from the main diagonal the matrix $M$ simply gets filled with a maximum distance value instead of the calculated distance.

## 4 APPLYING ALGORITHM TO TEST DATA SETS

### Test data set Athens

The first example is the test data set „Athens“ from the previous paper. The input stereo pair can be seen in fig. 4.

Figure 4. Stereo pair „Athens“

Applying the depicted DTW/dynamic programming algorithm with a window size of 3 to the stereo pair from fig. 4 yields fig. 5, left. As can be seen there are some blunders. Therefore after calculating the DEM a vertical median filter is applied to reduce this blunders significantly yielding fig. 5, right. All further DEMs are calculated with this vertical median and a window size of three unless otherweise noted.

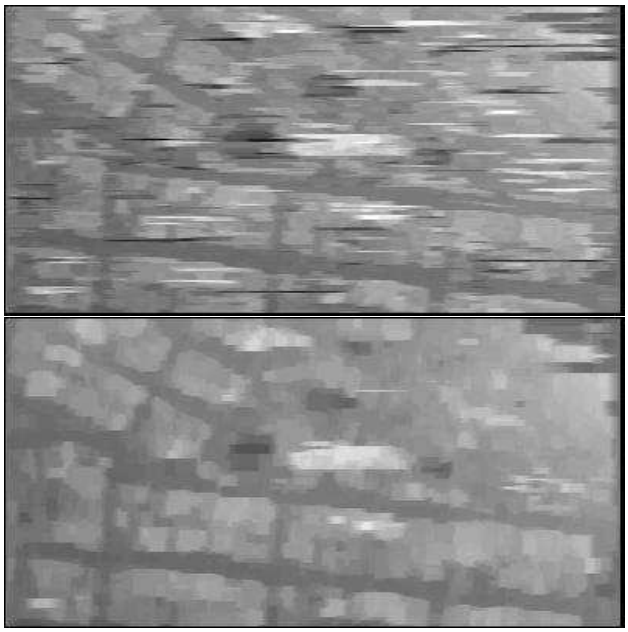

Figure 5. Applying dynamic programming algorithm to stereo pair of fig. 4 results in the DEM shown left. Applying a median filter reduces blunders significantly (right).

As can be seen the image structure is represented very well by the DEM. In the upper right corner erroneous stripes can be found at the right border. These errors follow from the relative height of the hill in this part of the image and the therefore missing correlating gray value profiles by reason of the border of the image. These border effects vanish if sufficient wide input images are given as can be seen at the top of fig. 19.

Taking into account not only the distance of the two gray values in one point of the two corresponding epipolar lines but in fact small windows in row and column direction around the demanded point gives much better results.

Calculating the DEM with varying window sizes is shown in fig. 6. An adequate compromise between blunders and smearing to large areas can be found at a window size of about three. This means that for calculating the distance between two points on an epipolar line not only the gray value distance of the two points but rather the sum of this and the point by point distances of all surrounding points should be used. A window size of one represents only the point by point distance.

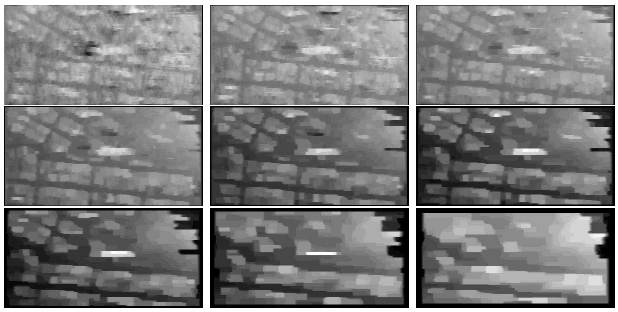

Figure 6. Results of the algorithm with window sizes 1, 2, 3; 4, 6, 8; 10, 15 and 20 pixel

To proove if the algorithm leads to consistent results it was applied to a image pyramid as shown in fig. 7. As can be seen the results are consistent with the reduced resolution and lead to mean heights of larger areas. The parallaxes shrink in the same degree as the

images get smaller. Thus, there remain in the last image only two height levels since the method uses only full integer pixel parallaxes.

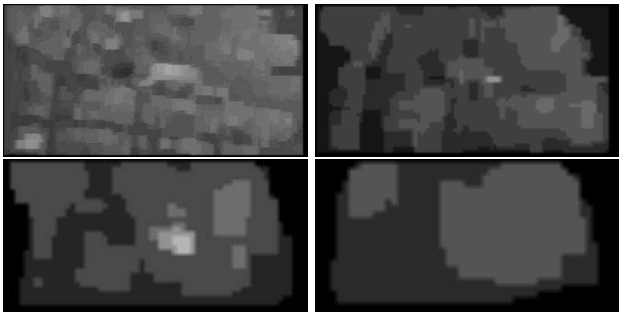

Figure 7. Algorithm applied to image pyramid (2, 4, 6, 8 times smaller)

Since the shown DTW algorithm is based on similar assumptions as the first experiments in the previous paper of generating a high resolution DEM from VHR satellite data a comparison of the methods is given in fig. 8.

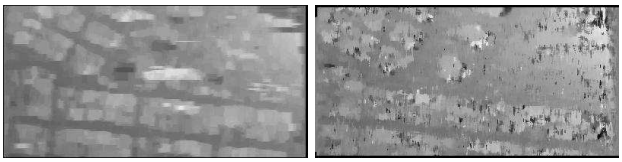

Figure 8. Comparison of the resulting DEMs from the „Athen“-scene of fig. 4 created with the DTW algorithm (left) and the previous algorithm (right)

Comparing the two resulting DEMs reveal noticeably fewer blunders in the DEM produced with the (new) DTW algorithm (left) in comparison to the DEM from the (old) column algorithm. In the same way a smaller horzontal blur can be observed. A few larger erroneous areas in the DTW-DEM can be seen which will be discussed later.

For comparing the results with a DEM generated by image matching after the method described in Lehner and Gill (1992) see Krauß et al. (2005).

**Test data set Tsukuba**

The second experiment is done with the test data set „Tsukuba“ as described in Scharstein and Szeliski (2002). Scharstein and Szeliski provide together with the epipolar stereo images for this data set also a real depth map on http://cat.middlebury.edu/stereo. This depth map is shown together with the stereo image pair in fig. 9.

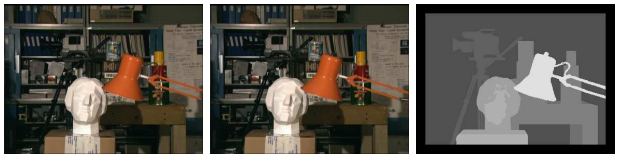

Figure 9. Stereo pair „Tsukuba“ and associating true depth map from Scharstein and Szeliski (2002)

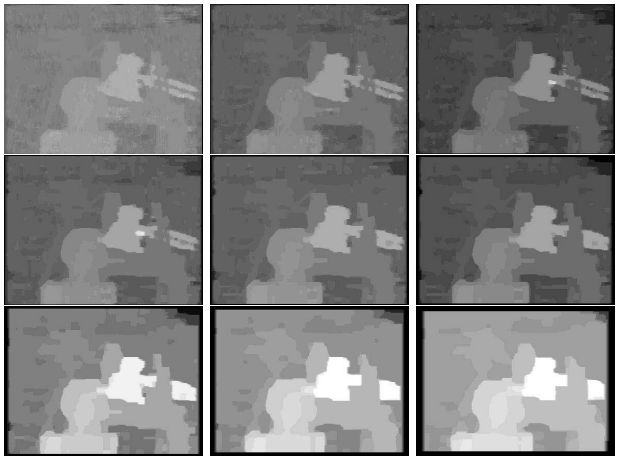

Figure 10. Results of the algorithm with window sizes 1, 2, 3; 4, 6, 8; 10, 15 and 20 pixel

Applying the DTW algorithm to this test data set yields – depending on the window sizes used – the DEMs shown in fig. 10. In almost the same manner as discussed with the previous example a window size of three is the best trade off between blunders and a loss of resolution.

Comparison of the resulting DEM with the DEM created by the „column“ algorithm described in the previous paper is shown in fig. 11. As can be seen the results are significantly better with the new DTW algorithm in comparison to the previous „column“ algorithm.

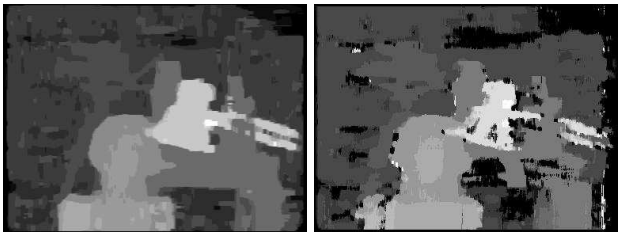

Figure 11. Comparison of the resulting DEMs from the „Tsukuba“-scene of fig. 9 created with the DTW algorithm (left) and the column algorithm

## 5 DISCUSSION

Due to the existence of a „true depth map“ for the test scene Tsukuba a direct comparison of this true depth with the calculated DEM by the DTW algorithm is possible. In figure 12 three profiles A, B and C are marked. These profiles are shown in figures 13 through 15 – the true depth map in red and the calculated DEM in green. As can be seen in all three profiles the correspondence between the true depth and the calculated DEM is pretty good. Actually the calculated DEM is a little bit blurred and starts in many cases a bit left of the true depth.

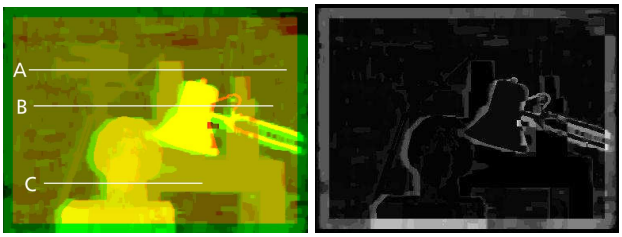


Figure 12. Discussed profiles A, B and C in the true DEM and the calculated DEM (left) and difference image of true and calculated DEM (right)

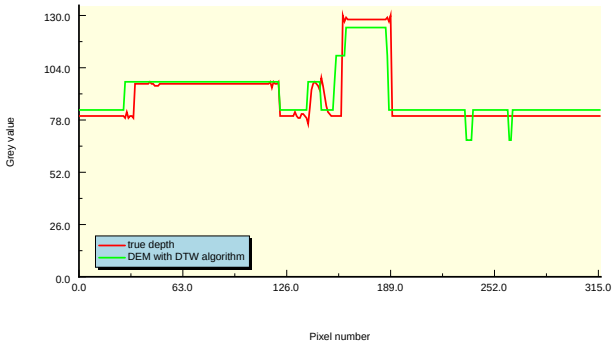


Figure 13. Profile A, crossing the bookshelf, the camera and a can in front

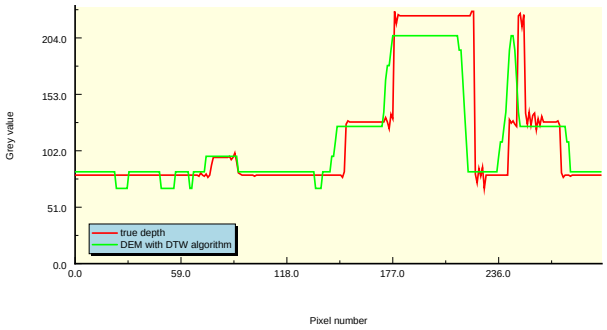


Figure 14. Profile B, crossing the tripod, books and cans on the table and the lamp (with cable!) in front

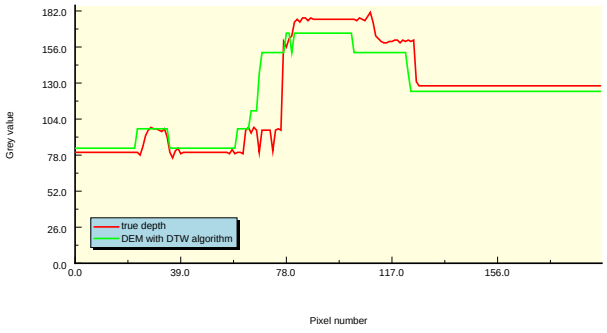


Figure 15. Profile C, crossing tripod, statue and table

At present the algorithm works only with parallaxes of full pixels – no subpixel refinement is done. The calculated DEM covers in this example only displacements of 0 up to 24 pixels. For better perceptibility these are mapped to the full range of 255 gray levels. That means a step of about 10 in the gray value of fig. 13 to 15 is one pixel in parallax direction.

Calculating the difference of the true depth map and the calculated DEM results in fig. 12, right. This difference shows a standard deviation in gray values of about $\sigma = 23.7$ which resembles about 2 pixel in parallax.

Unfortunatelly there is no ground truth (reference height model) available for the Athens data. Extrapolating the results found above and assuming a comparable behavior with both test images will lead to a standard deviation in height by about 2.7 m in the Athens-scene (angle of convergence = $2 \times 20^\circ$, resolution of Ikonos about 1 m).

Applying the algorithm to a larger area from the Athens dataset yields fig. 19. Beside the terrain structure (hill on the top of the image, depression in the lower left part) especially streets and buidings can be clearly identified. Even high vegetation like trees are found by the algorithm as can be seen on the slopes of the hill in the top of the image.

### Incorrect calculated areas

Analysing the resulting DEM from the Athens scene there are some regions which are obviously calculated wrong. The largest problem area is shown in fig. 16. This area covers a multi-story building with a big unstructured rooftop.

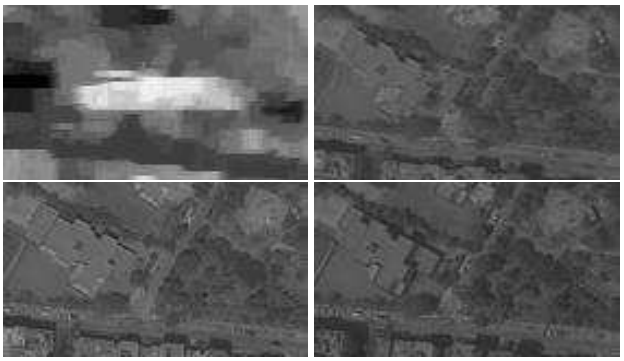

Figure 16. First incorrect calculated area (DTM, ortho image; left and right stereo; window size x and y: 3)

Experiments with changing window sizes and the assumption of too little structure in this area leading to errors doesn't show significant improvement. In fig. 17 calculation of the DEM with varying window sizes in horizontal direction and in fig. 18 varying window sizes in vertical direction are shown.

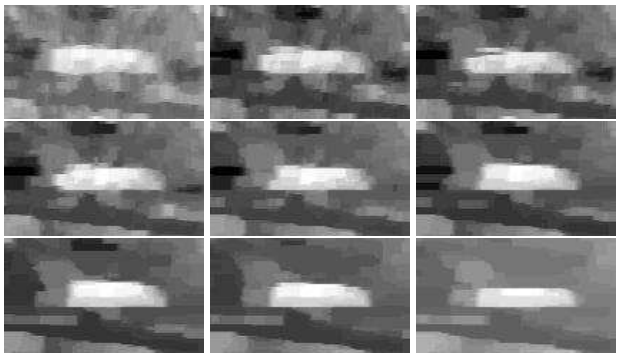

Figure 17. Area of fig. 16 calculated with changed window size in x direction (1, 2, 3; 4, 6, 8; 10, 15, 20)

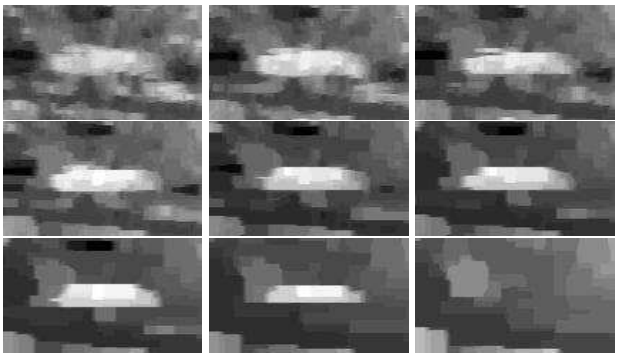

Figure 18. Area of fig. 16 calculated with changed window size in y direction (1, 2, 3; 4, 6, 8; 10, 15, 20)

Until now no reason for this kind of errors can be found and further investigation for eliminating these in fact rarely occuring errors will be necessary.

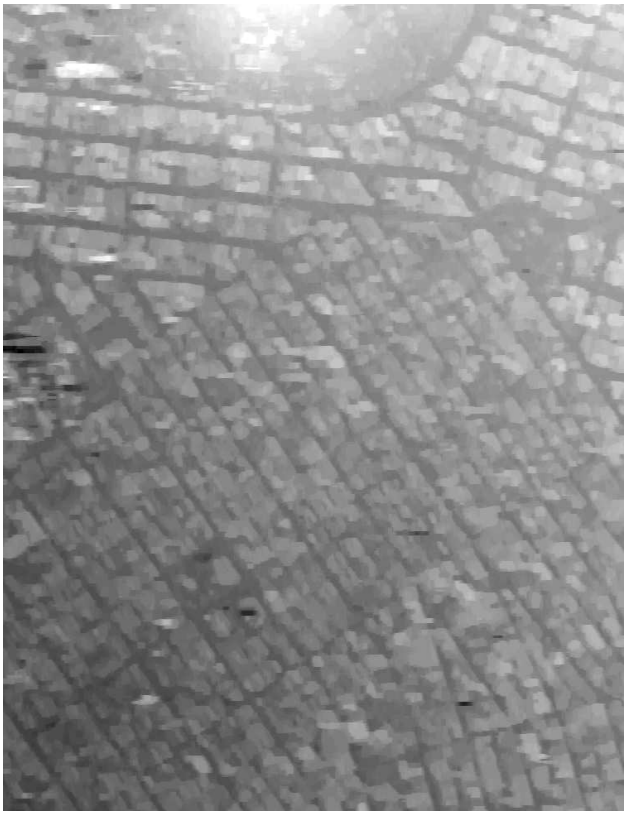

Figure 19. DEM of a larger area out of the Ikonos image Athens. fig. 4 can be found in the upper left corner. Urban structure is fairly good observable to distinguish between streets and build up objects (or high vegetation like trees)

## 6 CONCLUSION AND OUTLOOK

The presented DTW/dynamic programming algorithm produces very promising results in the derivation of DEMs from VHR stereo data from space. It behaves significantly better than other algorithms applied in earlier work. Especially for highly structured scenes like urban area its main potential is showing up.

The streets and house blocks can be seen clearly in the DEM but the absolute derived heights have still to be cross checked with reference data. Furthermore the reason for the generation of clearly erroneous areas has to be analysed and measures to be developed to reduce them significantly. For an improvement of the height accuracy sub-pixel calculations will be included in the process.

## REFERENCES


Baltsavias, E. P., 1997. Matching Verfahren und automatische DTM Generierung. Technical report, ETH Zürich

Culjat, D., 1999. Dynamische Programmierung in der Spracherkennung. Technical report, FU Berlin, FB Informatik

Krauß, T., Reinartz, P., Lehner, M., Schroeder, M. and Stilla, U., 2005. DEM generation from very high resolution stereo data in urban areas. ISPRS 34 (8)

Lehner, M. and Gill, R., 1992. Semi-automatic derivation of digital elevation models from stereoscopic 3-line scanner data. ISPRS 29 (B4), pp. 68–75

Scharstein, D. and Szeliski, R., 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. IJCV 47(1/2/3), pp. 7–42

Schenk, T., 2004. From point-based to feature-based aerial triangulation. ISPRS Journal of Photogrammetry and Remote Sensing

Schiele, B., 2005. Human Computer Systems. Technical report, TU Darmstadt