LEAST-SQUARES MATCHING USING ELLIPTICAL AREAS: RESULTS, ACCURACY, ADVANTAGES AND DISADVANTAGES.

D. Skarlatos *, A. Georgopoulos

NTUA, School of Rural & Surveying, Laboratory of Photogrammtry, 9 Iroon Polytexniou, Athens 15780, Greece, (dskarlat,drag)@central.ntua.gr

Commission III

KEY WORDS: DTM/DEM, quality, reliability, algorithms, automation, accuracy.

ABSTRACT:

The use of a square template in least squares image matching has been established as a well known and tested method. This paper suggests the use of an elliptical template as alternative to the square one. The main stimulus for the adoption of the ellipse was the disturbing shift of the matched point along linear features. When the square template was used around a point in a linear feature, the matched point was not in the correct position, due to slipping along that feature. This phenomenon is expected in least squares matching (LSQM) because all points along the line are similar and hardly distinguishable. Even the use of epipolar geometry cannot solve the ambiguity of this problem, when the linear features are parallel or almost parallel to the epipolar line. The use of a dynamic ellipse oriented with the large axis along the linear feature uses considerably more information, thus strengthening the matching process along that direction. The necessary information about the size, shape and orientation of the ellipse are dynamically calculated based on local image content and data from the previous iteration.

1. INTRODUCTION

Image matching is a very important aspect of really digital photogrammetry since it supports automation. Feature-based and area-based matching are the main categories in image matching, with the latter being preferred by surveyors and photogrammetrists due to increased accuracy and possibility of statistical analysis of errors.

Least square template matching, as an extension to simple correlation coefficient matching, was first introduced in 1984 (Gruen et al). The method has gone a long way since then, was enhanced using epipolar geometry (Gruen, Baltsavias, 1985), multiphoto image matching (Baltsavias, 1991), simultaneous multi point matching (Rosenholm 1986), pyramids (Baltsavias,1988), combination of feature-based and area-based matching (Forstner 1986), etc. Many enhancements over the basic model have been introduced since 1984. The versatility of least squares and their ability to combine data from any source, provided that the mathematical model is well defined, has boosted research. Numerous variations of image matching using least squares have been published the last years, using the basic least squares matching with 6 geometric unknowns (or 4 unknowns in case similarity was adopted) as their core module. All these approaches were based on square templates, a decision probably based to pixel layout and limited computer power.

Since the early 90s, when image matching research was at its peak, there has been little algorithmic enhancement. Photogrammetric research has shifted to automatic aerial triangulation and digital aerial cameras, leaving image matching to signal analysts, and machine vision as part of artificial intelligence. Electronic engineers are interested in pattern recognition while accuracy isn't their priority.

This research is being done as part of a PhD thesis concerning image matching in general. Under this prism the effort was concentrated for the production of a stand-alone matching algorithm with the following characteristics:

- Independence from external data. The fact that the algorithm should be used for automatic relative orientation and for the new digital cameras, which depend on line arrays and therefore epipolar geometry does not apply as we know it, were the main aspects that lead to the adaptation of this approach.
- Reliability. In this way automatic relative orientation could depend on these points and the matching could be used for the extraction of a reliable Digital Elevation Model (DEM), where extensive manual corrections would not be necessary anymore. It is not to forget that the time needed for DEMs automatic collection and necessary correction is almost comparable to the manual collection. Actually many commercial companies still use manual collection in certain cases although they own software packages with automatic (DEM) collection.
- Self adaptation. LSQM has a vast number of parameters, which should be adjusted for each model. In high end commercial systems the number of parameters is huge and only matching experts can set them correctly. It is not rare to find application engineers of a specific software package, who cannot set the parameters correctly. Even when set correctly, these parameters do not comply with features all over the model area, and therefore the matching algorithm should be "clever" enough to adjust itself to the content and the information around each point. This not only helps inexperienced users, who cannot define the parameters, but also ensures the best combination of parameters in each point. Hence the algorithm was designed so that almost all parameters are self adaptive through internal procedures. A large effort has been made to keep manually set parameters to a minimum.

DEM generation and relative orientation are the most demanding image matching applications and that is why research was focused on these topics.

In general the procedure for DEM generation looks like this:

1. Pre-processing of the images (image enhancement)
2. Initial approximation (usually using feature based matching, pyramids, or neighbourhood techniques)
3. Least Squares Matching (LSQM)
4. Filtering of blunders in 3D or 2D space

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

5. Generation of the surface which is done either by regularisation using interpolation on a given grid or by Triangulated Irregular Network (TIN)

Therefore the final DEM accuracy depends on all these factors. The most recent developments in automatic DEM generation are focused on pre and post processing of data.

It was the authors' belief that if the main matching algorithm was robust enough, the filtering of blunders would become simpler or even useless. Therefore the research was focused in the core of the matching algorithm, the LSQM.

In the initial efforts a number of existing techniques were tested. From the early stages it was noticed that points on lines or edges could not be properly matched. It is usual, for points along linear features to be wrongly matched, the matching ending due to iteration limit, without returning a valid solution. During the error analysis of the LSQM the error ellipse from the variance-covariance matrix (Qxx) was drawn over points, which were wrongly matched. That was when the idea of using the ellipse as a matching template was born. It seemed promising because the matching would use more information along the edge, where the localisation is ambiguous. Since the localisation is good along the direction perpendicular to the line, one can afford to use less pixels along that direction in favour of more pixels along the linear feature.

2. METHODOLOGY

Although as a concept it is rather simple, a number of questions rise when trying to implement this idea in the computer. For this particular research the left image is being considered as the template (also found as master in bibliography) and the right image is the search area (also found as slave). The main concept is that the Q_{XX} matrix should be used to formulate an ellipse, which is going to be used for pixel selection for matching. Because of this there are two basic implementation differences from the standard LSQM procedure.

- The first one is that the template is not fixed during the process. Pixels used for matching depend on the shape and orientation of the ellipse formulated on the search image.
- The second one, which actually originates from the previous one, is that the interpolation takes place over the pixels on the template image rather than on the search image.

The schematic procedure of the method can be seen in fig. 1.

From the Qxx matrix define the main parameters of the ellipse (two axes and one angle), which correspond on the right image.

Since the corrections over the 6 parameters of the affine correspond to the deformed template on the search image, it is quite clear that the Qxx matrix corresponds on the right image.

The formulae are quite straightforward and can be found in any least squares textbook. In this particular case formulae are from Balodimou (2000). The characteristic values for an ellipse are shown in figure 2.

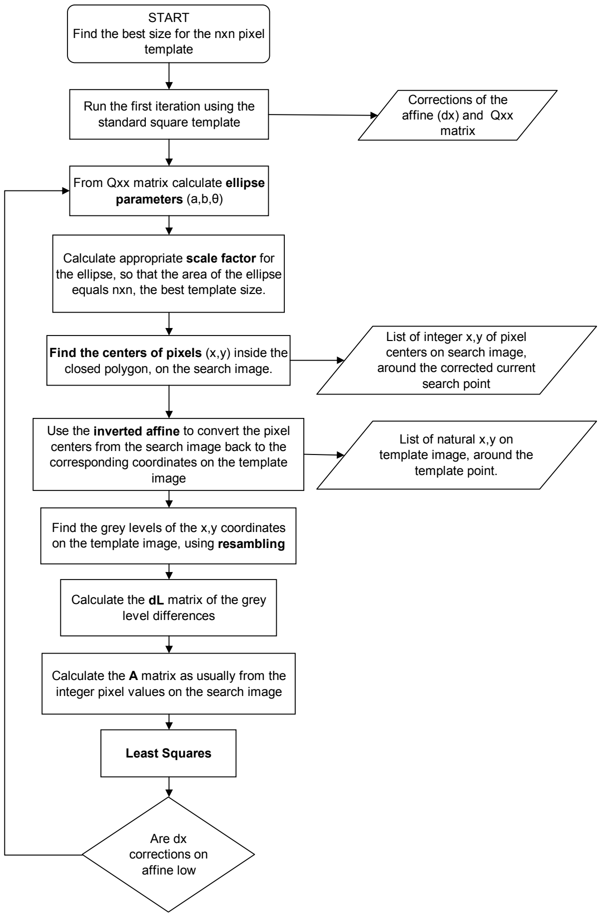

Figure 1. Flow chart of the implemented method.

$$\tan(2\;) = \frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2} \quad \text{or} \quad \tan(2\alpha) = \frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2} \tag{1}$$

Supposing θ_{max}, θ_{min} and α_{max}, α_{min} are directions for the major and the minor semi-axis:

$\sigma_x > \sigma_y$	$\sigma_{xy} > 0$	$\theta_{max} = \theta$	$\alpha_{max} = \alpha + 100^g$
$\sigma_x > \sigma_y$	$\sigma_{xy} < 0$	$\theta_{max} = \theta + 200^g$	$\alpha_{max} = \alpha + 100^g$
$\sigma_x < \sigma_y$	$\sigma_{xy} > 0$	$\theta_{max} = \theta + 100^g$	$\alpha_{max} = \alpha$
$\sigma_x < \sigma_y$	$\sigma_{xy} < 0$	$\theta_{max} = \theta + 100^g$	$\alpha_{max} = \alpha + 200^g$
$\sigma_x = \sigma_y$	$\sigma_{xy} > 0$	$\theta_{max} = 50^g$	$\alpha_{max} = 50^g$
$\sigma_x = \sigma_y$	$\sigma_{xy} < 0$	$\theta_{max} = 150^g$	$\alpha_{max} = 100^g$

The magnitude of the two main semi axes are calculated from the following formulae:

$$\sigma_{max}^2 = \sigma_u^2 = \frac{1}{2}\left[\left(\sigma_x^2 + \sigma_y^2\right) + \sqrt{\left(\sigma_x^2 - \sigma_y^2\right) + 4\sigma_{xy}^2}\right] \tag{2}$$

$$\sigma_{min}^2 = \sigma_v^2 = \frac{1}{2}\left[\left(\sigma_x^2 + \sigma_y^2\right) - \sqrt{\left(\sigma_x^2 - \sigma_y^2\right) + 4\sigma_{xy}^2}\right] \tag{3}$$

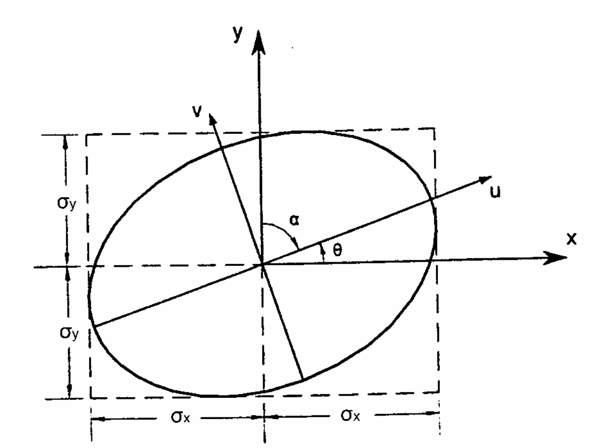

Figure 2. Ellipse characteristics.

The coverage of the calculated ellipse is a very small area. This is expected considering that the final ambiguities along the two main directions are very small, at the magnitude of ±0.1 pixels or even lower. Hence a scale factor must be applied in the two main axes. In fact the information for the best size of the square template is already available from a previous step of the algorithm, described in Skarlatos, 2000. An algorithm is applied prior to matching to decide about the best possible size of the square template. The decision is based on statistical values about information around the pixel in the left (template) image. It should be noted here that this algorithm is location invariant and investigates each patch size based on the square template concept, not on the ellipse itself. It is not repeated during iterations, instead it is applied once prior to matching in each point.
It is possible though to use the existing self-adaptive template algorithm to recalculate the best size of the ellipse, based on its shape and orientation. The constraint is that the checking should be done for selected areas of 80,100,150…900 pixels (equivalent to 9x9-31x31 window size). Recalculations of the description of the ellipse and the pixels within cause unacceptable delay and therefore such modification was rejected.

Find appropriate scale factor for the ellipse.
The area of the ellipse is πab, where a and b are the main semi-axes, or in this case $\pi \cdot \sigma_{min} \cdot \sigma_{max}$. Therefore the scale factor for each semi-axis is $\sqrt{\frac{new\_area}{\pi \cdot \sigma_{min} \cdot \sigma_{max}}}$. If each semiaxis is multiplied by this factor, the new ellipse has area equal to new_area. Proportions and orientation of the ellipse are maintained, absolutely necessary to the concept of this algorithm.

Find the pixels in the ellipse.
Theoretically pixels belonging to the ellipse should have more than 50% of their area in it. This method of pixel identification consumes a lot of computer power, therefore a simpler method was used. If the center of the pixel is inside the ellipse then the pixel belongs to the ellipse.
Therefore the two focal points of the ellipse are calculated e_1 and e_2 (fig. 3). The focal points are located on the large axis at distance $\gamma = \sqrt{\sigma_{max}^2 - \sigma_{min}^2}$ from the centre of the ellipse, hence their coordinates on the local coordinate system of the ellipse are $(\pm\gamma,0)$. By applying a rotation angle θ and two shifts X_o and Y_o, (X_o,Y_o) being the centre of the ellipse, the coordinates are transformed in the image coordinate system

$$\begin{bmatrix} X \\ Y \end{bmatrix}_{e_1} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{e_1} + \begin{bmatrix} X_o \\ Y_o \end{bmatrix}_{ellipse\_centre} \tag{4}$$

For a point (center of pixel) to be inside the ellipse it is necessary:

$$dist(e_1,p) + dist(e_2,p) < 2a \tag{5}$$

Where a is the big semi-axis and p the pixel under investigation.
This check is being done on all pixels within a square with sides of 2a, ensuring all possibilities for the direction of the ellipse are included in the check. This check is simple and fast. The only drawback is that the number of finally selected pixels does not coincide exactly with the desired area of the ellipse, as calculated on step 2. Statistically this is less than 2% for the 99% of the cases. For small ellipses this percentage may go up to 3%, but drops rapidly when size increase, and therefore returns the aforementioned results over a matched model. In any case such discrepancies do not affect the general idea of the proposed method.

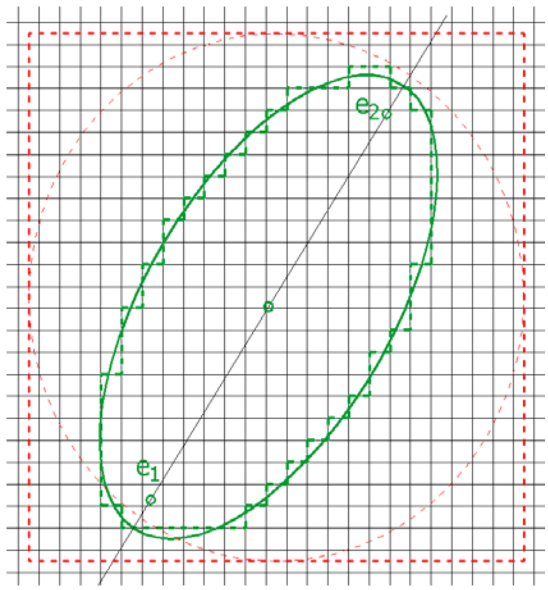

Figure 3. Ellipse (green solid line) with check area (red dashed), finally selected pixels (green dashed). Expected area 221 (=11x11) and finally selected pixel 218, representing differentiation less than 1.5%.

Use the inverse affine transformation to locate these pixels on the left image
In order to perform LSQM the same pixels should also be located on the left (template) image. In order to do so the inverse affine transformation from the previous iteration is used to find the co-ordinates of these pixels on the left image. It is expected that after the transformation the left pixels will be in random positions (not integer values) and therefore interpolation is necessary to find the grey level values for these positions. The values are used as floating point numbers for further calculations.

Formulate the matrices, A and l for least squares and solve them for the 8 parameters

The model used for LSQ, adopts the affine transformation for geometric corrections with two additional parameters for radiometric corrections and is identical to the model described extensively in Baltsavias 1991 and Gruen 1996. Supposing that the geometric transformation is:

$$
\begin{aligned}
x &= a_{11} + a_{12}x_o + a_{21}y_o \\
y &= b_{11} + b_{12}x_o + b_{21}y_o
\end{aligned}
\qquad (6)
$$

where the unknowns are

$x^T = \{da_{11}, da_{12}, da_{21}, db_{11}, db_{12}, db_{21}, r_s, r\}$

The main equation for every observation (grey level difference between right and left interpolated pixel) which forms the A and l matrix:

$$
f(x,y) - e(x,y) = \overset{o}{g}(x,y) + g_x da_{11} + g_x x_o da_{12} + g_x y_o da_{21} + g_y db_{11} + g_y x_o db_{12} + g_y y_o db_{21} + r_s + \overset{o}{g}(x,y) r_t \qquad (7)
$$

where

$g_x = \dfrac{\partial \overset{o}{g}(x,y)}{\partial x}, \; g_y = \dfrac{\partial \overset{o}{g}(x,y)}{\partial y}$

are the partial derivatives along x and y axis respectively.
The matrix equation formed is $Ax = l$ and the solution being

$$
x = (A^T A)^{-1} A^T l \qquad (8).
$$

If the corrections on dx and dy are still high, go to step 1

3. APPLICATION OF ELLIPSE MATCHING AND COMPARISON WITH THE STANDARD SQUARE TEMPLATE.

The main algorithm of the method is described in detail above. At this time the matching software including the algorithm is used as a learning tool for customisation and optimisation. Hence there is a big number of parameters that can be adapted or self-adapted. For simplicity and comparison reasons it should be mentioned that both algorithms are tested using

- Maximum template size 41x41 pixels. This means that the automatic template size algorithm will check all templates between 7x7 and 41x41 to find the best size for a square template. If no template size is considered good enough for matching, then the matching in this position is being done using the maximum allowed (41x41 in this case). Otherwise the matching will be done using the template found. If matching with this template fails then the matching will be attempted again with a bigger template, actually the next template size will be

  $$
  nexttempl = currenttempl + \frac{maxtempl - currenttempl}{2} \qquad (9)
  $$

  This continues in case of failure until the maximum template defined by the user is reached. A template of 41x41 is rather big, but even so in some cases of homogeneous background it is useful. Of course both methods start with the same template.
- The iterations stop if both dx and dy corrections are lower than 0.2 pixels or if their number exceeds 12.
- Both methods use 8 unknowns, 6 geometric and 2 radiometric parameters.
- Correlation is being done prior to matching so that the matching has initial approximations better than 1-2 pixels, which is the convergence radius for the LSQ method. Since this technique is applied here, the starting pixel (initial approximation) for both methods is the same.

In order to test the initial motivation and the theoretical background for the ellipse method four examples will be presented, all for points along linear features.

3.1 Case 1

This case is described to demonstrate that the square template may return a "correct" match in an erroneous position, while elliptical template returns the correct position.
In both cases the best template size was found to be 13x13 pixels, since the algorithm is invariant of the LSQM which follows. Both methods start from the same initial approximation (pixel in the right/search image), since this point is provided using correlation.
Both methods return a "correctly" matched point. As shown in figure 3, the square template method returned a wrongly matched point, due to the aforementioned shift, which occurs, in linear features. This phenomenon is particularly interesting here, because the square template fails although it has rich information (the dense shadow) just 3 pixels away. After failing to match the square template of 13x13 pixels, the algorithm used a 29x29 template, which returned after 3 iterations a matched point, which is obviously wrong (fig. 4).
The ellipse using 169 pixels (equivalent to 13x13) returned a correct match after 3 iterations. The fact that the ellipse is more accurate than the square is verified from the σ_o for the gray level differences. In the square method σ_o is 25.14 while the ellipse returned a much smaller σ_o of 12.73, indicating that the match of the ellipse was much stronger.
It should be mentioned that due to the simplifications made for the ellipse, in terms of shape and pixels used, the final number of used pixels for matching was 169 and 167 in the second and third iterations respectively. A deviation of 2 pixels in 169 pixels, or 1.2% is considered negligible and certainly unable to affect the final result.
It should also be mentioned that in this case the ellipse method was faster than the square one, not to mention that if the algorithm was used with 13x13 template instead of the self-adaptive, the square would have failed completely.

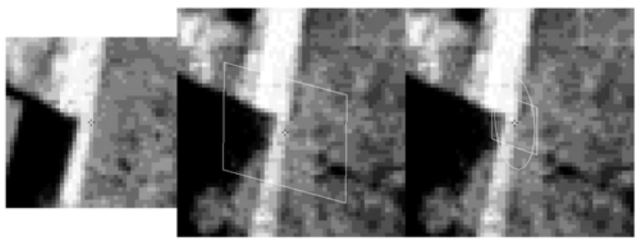

Figure 4. Comparison between the square and ellipse template:Case1. From left to right: The left/template image, the matched point in the right (search) image using square, the matched point in

the right image using ellipse. The rectangle in ellipse method is displayed as a measure of the affine parameters.

3.2 Case 2

Here a 25x25 (=625) pixels template was found to be the best. Both methods started from the same approximation and returned a correctly matched point after 8 iterations. Once again the square template is obviously wrong, misplacing the matched point towards the centre of the pavement, probably because of the shadow. As one can see in figure 4, the pavement in this particular spot is under a tree, causing left and right images to differ considerably, not to mention a "strange" line on the left image due to scanning (fig. 5).
On the other hand the ellipse method provides a much better localisation, exactly on the edge of the pavement. During the iterations, the maximum and minimum of the pixels used were 634 and 624 respectively, instead of 625. These differences (1.4%) are considered minor and certainly not able to influence the final match. Once again, the ellipse returned a better value for σ_o, 15.25 against 21.12 of the square template, verifying initial considerations.

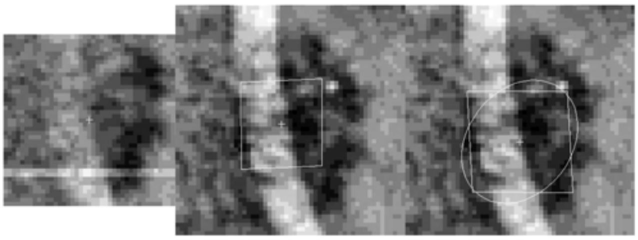

Figure 5. Comparison between the square and ellipse template:Case2. Bad localization of the square, returns wrong point. The line in the left (template) image is the effect of bad scanning, which surprisingly doesn't affect the match. The final affine parameters are different. Observe the shape of the rectangle in both cases.

3.3 Case 3

This is described to show that in some cases the square template cannot return a match, while elliptical template returns the correct position.
In this case the best template was found to be 29x29 pixels.
The square method failed completely. It did not return a matched point. After failing using the 29x29 template, it used a larger template of 37x37 to include more information, and after that a 41x41 template, which is the largest template allowed by the user. After the failure of the 41x41 template, which can be seen in figure 5, the matching algorithm returned a complete failure, instead of a point (fig. 6).
The ellipse method used the 29x29 or 841 pixels and found a correct match after 3 iterations. Actually in the second and third repetition 840 and 846 pixels were used instead of the expected, 841, but this is also considered a small deviation since it is 0.6%.
In this particular case the ellipse method not only did find a correct match accurately, but it was faster than the square.
In this case σ_o was 7.81, the lowest from all presented cases, although this case is obviously the weakest. This can be explained by the fact that the two images are similar and therefore the grey level differences are very small while the algorithm cannot find a strong solution, because the information around the pixel is the same for every point on the line.

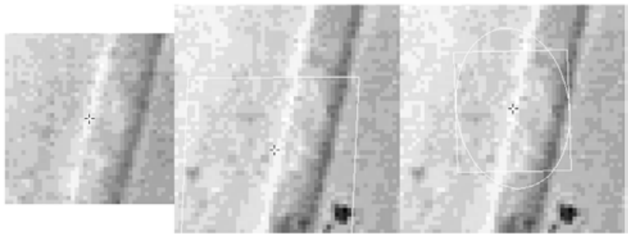

Figure 6. Comparison between the square and ellipse template:Case3. Complete failure of the square template, even with the largest 41x41 template. The shift along the line is clear.

3.4 Case 4

This case is described to show that the ellipse works just as well or even better as the square template in normal cases.
The best template was found to be 15x15. Both methods return a correct match after 2 iterations (fig. 7). Elliptical template uses 223 pixels instead of 225, which represents a 0.9% decrease of the total pixels used. This difference is incapable to affect the final match.
There is a difference between the returned values of the matched point, 356.79, 898.55 (ellipse) and 356.58, 898.42 (square). The difference of 0.19, 0.13 pixels, which is almost indistinguishable in figure 6, is justified if one considers that the expected accuracy of LSM is 0.1-0.2 of the pixel (Guelch E.,1988). The same figure of 0.2 pixels for random points is also reported in Trinder, J.C. et al.The σ_o, which is a measure of precision of the final match, is in favor of the ellipse (7.32 against 10.54 of the square), thus indicating that the elliptical template might return a more accurate position.

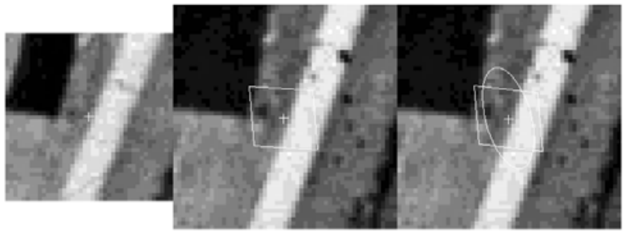

Figure 7. Comparison between the square and ellipse template:Case4. Both cases return a corrct match.

4. CONCLUSIONS AND FURTHER RESEARCH.

Until know ellipse has been tested against square template, manually over a number of features, including points, corners, uniform areas etc. In point features and in homogeneous areas, the ellipse is almost a circle. In all cases ellipse returns a better σ_o value, which is indicative of better precision. In certain cases, especially in linear features, it provides not only more accurate results, but also correct results even in some cases where square fails completely. The superiority of the elliptical template is shown numerically in table 1.

	Case 1		Case 2		Case 3		Case 4	
	Sq.	Ell.	Sq.	Ell.	Sq.	Ell.	Sq.	Ell.
Template sizes used *(the last one is the one which returns the point)*	13x13 29x29	13x13	25x25	25x25	29x29 37x37 41x41	29x29	15x15	15x15
Pixel decrease percentage	-	1.2%	-	1.4%	-	0.6%	-	0.9%
Iterations on the last template used	3	3	8	8	-	3	2	2
σ_o	25.14	12.73	21.12	15.25	-	7.81	10.54	7.32
Returned point	YES	YES	YES	YES	NO	YES	YES	YES
Correct match	NO	YES	NO	YES	NO	YES	YES	YES

Table 1. Comparison of results.

The only problem that arises from the application of the proposed method is the complication of the calculations, but then again this is the only way to attain better results. Complication of calculations leads to more computer time. The algorithm has not been timed since it has been used until now manually and time differences cannot be observed, but it is predicted that it will be slower than the square.

It should be mentioned though that not much attention should be given in speed, because computer power doubles every 1.5 years. When LSQM was first introduced it was very slow for the contemporary computers, not to mention the quality of CCD sensors. Today, matching over a whole model, producing 18000 points can be completed in 3 minutes and for 2,5 million points in 30 minutes in an average computer.

The problem might be evident when applied in DEM collection. Prior to this code optimization will decrease the algorithm's speed by half. Use of fewer vertices to describe the ellipse is another possible source of time saving. Reduction by 20%, will save time almost 15% over the whole matching algorithm. Another interesting feature is that after the second iteration the ellipse does not change considerably both is shape and orientation, meaning that it is almost useless to reform it after each iteration. This is another point where processing time can be saved.

In 2 cases (1 and 3), the square method fails in the suggested template and uses a bigger one in order to find a solution. In the same cases ellipse uses the suggested template and is equal or faster, in terms of total iterations, in all cases. Therefore the ellipse method compensates speed, up to a point by itself.

Epipolar geometry provides a very good solution for linear features nearly perpendicular to the epipolar line (Baltsavias, E., 1993). The ellipse provides solution for all linear features, without the use of relative orientation, and therefore is a universal method, while being more accurate for all points. Failure rate, including points returned from square template as correct, without being so, is reduced considerably. Hence it is safe to conclude that it is a promising method which requires further research. The next step is speed optimisation and application on DEM collection of different objects and scales to verify these conclusions. Comparison will be done against square template DEM as well as against a reference DEM.

References

Forstner, W., 1986, On feature based correspondence algorithm for image matching and least squares matching, *International Archives of Photogrammetry and Remote Sensing*, Vol. 26, P3, Rovaniemi, pp. 150-166.

Gruen, A.W., 1985, Adaptive least squares correlation, *South African Journal of Photogrammetry, Remote Sensing and Cartography,* 14(3):175-187.

Gruen, A., Baltsavias, E., 1985, Adaptive least squares correlation with geometrical constraints, *Proceedings Of SPIE (The Society of photo-Optical Instrumentation Engineers)*, Vol. 595, pp. 72-82.

Rosenholm, D., 1986, Accuracy improvement of digital matching for evaluation of digital terrain models, *International Archives of Photogrammetry and Remote Sensing*, Com III, Vol. 16 –3/2, pp. 573-587.

Rosenholm, D., 1987, Empirical investigation of optimal window size using the least squares image matching method, *Photogrammetria*, Vol. 42, pp. 113-125.

Baltsavias, E., 1991. *Multiphoto Geometrically Constrained Matching*. PhD thesis in Institute for geodesy and photogrammetry, Zurich, 221 p.

Skarlatos, D., 2000, Image matching towards maturity, *International Archives of Photogrammetry and Remote Sensing*, Amsterdam 2000, Vol. 33, Part B3, pp. 845-849.

Balodimou, A., M., 2000. *Theory of errors and position fixing, ambiguities in 2 and 3 dimensions*. Notes (in greek) from lectures in National Technical University of Athens. 24 p.

Atkinson, K., B., 1996. *Close range photogrammetry and machine vision*. Whittles Publishing, BristolScotland UK, 371 p.

Guelch, E., 1988, Results of test on image matching of ISPRS WG III/4. *International Archives of Photogrammetry and Remote Sensing*, Kyoto 1988, Vol. 27, Part B4, pp. 254-271.

Trinder, J., C., Becek, K., Donnelly, B., E., 1992, Precision of image matching, *International Archives of Photogrammetry and Remote Sensing*, Washington DC, Vol. 29, Part B3, pp 820,823

Baltsavias, E.P., Stallmann, D., 1993. SPOT stereo matching for digital terrain model generation. *Proceedings of 2nd Swiss Symposium on Pattern Recognition and Computer Vision*, Zurich, pp. 61-72.