USING SPANNING TREES FOR REDUCED COMPLEXITY IMAGE MOSAICING

Nikos Nikolaidis and Ioannis Pitas

Artificial Intelligence and Information Analysis Laboratory,
Department of Informatics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece
email: {nikolaid,pitas}@aiia.csd.auth.gr
web: http://poseidon.csd.auth.gr

ABSTRACT

Image mosaicing, i.e., reconstruction of an image from a set of overlapping sub-images, has numerous applications that include high resolution image acquisition of works of art. Unfortunately, optimal mosaicing has very large computational complexity that soon becomes prohibitive as the number of sub-images increases. In this paper, two methods which achieve significant computational savings by applying mosaicing in pairs of two sub-images at a time, without significant reconstruction losses, are proposed. Simulations are used to verify the computational efficiency and good performance in terms of matching error of the proposed techniques.

1. INTRODUCTION

Mosaicing is the process of reconstructing or re-stitching a single, continuous image from a set of overlapping sub-images. Image mosaicing is essential for the creation of high resolution digital images of architectural monuments and works of art (especially of those with considerable dimensions like frescoes and large-size paintings) for archival, digital analysis and restoration purposes [1],[2]. In such applications, the required very high resolution image acquisition stresses the limits of acquisition devices. To overcome this obstacle, digitization procedures that utilize the acquisition of different, overlapping, views, sometimes with the aid of sensor arrays, and positioning mechanisms are used. The acquired sub-images (Fig. 1 (a)) are subsequently processed by a mosaicing algorithm. If the field of view is split into M_1 rows of M_2 images each, it is easy to show that an M_1M_2-fold increase in resolution may be attained, compared to sensor resolution.

Mosaicing is also important in other areas that include the creation of high-resolution large-scale panoramas for virtual environments, image-based rendering, medical imaging [3], [8], aerial [9], satellite and underwater [10] imaging etc. Several image mosaicing techniques have been proposed in the literature [1]-[10].

The mosaicing process comprises of two steps. The first step involves the estimation of the optimal displacement of each sub-image with respect to each neighboring one (assuming only translational camera motion during the acquisition and no rotation or zooming). This is the most computationally intensive part of the entire process. In the general case of a set of $M_1 \times M_2$ sub-images, optimal mosaicing, i.e., mosaicing by concurrently searching for the optimal position of all sub-images, would require a search in a m-dimensional space, where $m = 2(2M_1M_2 - M_1 - M_2)$, the term in parenthesis representing the number of all pairs of neighboring sub-images. Obviously the computational cost becomes prohibitive, as the number of sub-images increases. The second step of the mosaicing process utilizes displacement information found in the previous step in order to combine each pair of neighboring sub-images with invisible seams and thus reconstruct the whole image.

In this paper we will focus on the first step and propose two methods for reducing the number of computations required to compute the sub-image displacements without affecting significantly the matching error. It is important to note that, despite the fact that the methods are illustrated for the particular case where sub-images are only displaced (translated) with respect to each other, the proposed matching methodology is applicable to more complex situations, e.g. cases that involve camera rotation or zooming when acquiring the sub-images.

2. MOSAICING OF TWO IMAGES

Before dealing with the general case of mosaicing an arbitrary number of sub-images, the case of two images will be studied, since it provides significant insight to the problem. In the following it is assumed that the displacement vector $\mathbf{d}$ is constrained to take values in the following set:

$$\mathbf{d} \in \{[d_1\ d_2]^T : d_i \in \{d_{i_{\min}}, \ldots, d_{i_{\max}}\},\ i = 1,2\} \qquad (1)$$

If $I_j(\mathbf{n})$, $j = 1, 2$, denotes the intensity of the j-th image at pixel coordinates $\mathbf{n} = [n_1\ n_2]^T \in W(\mathbf{d})$, where $W(\mathbf{d})$ denotes the area where the two images overlap (Fig. 2), then the matching error $E(\mathbf{d})$ associated with a specific displacement $\mathbf{d}$, can be expressed as follows:

$$E(\mathbf{d}) = \frac{\sum_{\mathbf{n} \in W(\mathbf{d})} \left| I_1(\mathbf{n}) - I_2(\mathbf{n}) \right|^p}{||W(\mathbf{d})||} \qquad (2)$$

where $||W(\mathbf{d})||$ denotes the number of pixels in the overlap area $W(\mathbf{d})$. For $p = 1, 2$ (2) expresses the Matching Mean Absolute Error MMAE and the Matching Mean Square Error MMSE, respectively. Subsequently, the optimal displacement value $\mathbf{d}_{opt}$ can be estimated through the following minimization:

$$\mathbf{d}_{opt} = \arg\min_{\mathbf{d}} E(\mathbf{d}) \qquad (3)$$

From (1) and (3) it is obvious that this minimization process requires the evaluation of (2) over all possible values of $\mathbf{d}$. Block matching techniques (2-d logarithmic search, three-point search, conjugate gradient search [11]) can be employed in order to reduce the computational cost associated with the exhaustive minimization procedure (3). These procedures may provide estimates $\hat{\mathbf{d}}_{opt}$ of the optimal displacement value $\mathbf{d}_{opt}$. In our simulations, the 2-d logarithmic search was employed.

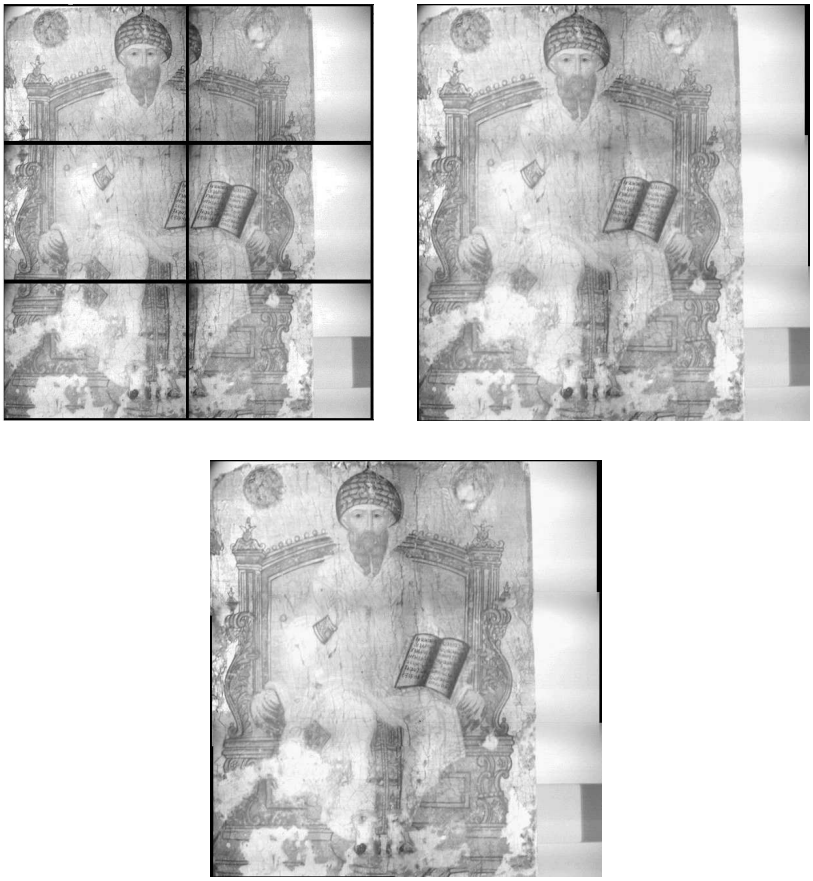

Figure 1: (a) A $M_1 = 3$ by $M_2 = 2$ sub-image acquisition of a painting. (b) Spanning tree mosaicing (STM) reconstructed image. (c) Reconstructed image after sub-graph spanning tree mosaicing (SGSTM).

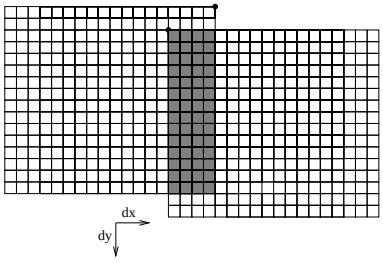

Figure 2: Two neighboring sub-images and the associated overlap region $W(\mathbf{d})$ (which is identified by the gray area) for a displacement of $\mathbf{d} = [-4\ 2]^T$.

3. MULTIPLE IMAGE SPANNING TREE MOSAICING

If $M_1 \times M_2$ sub-images are to be mosaiced, a displacement matrix $\mathbf{D}$ is involved. This matrix plays a role similar to that of the displacement vector $\mathbf{d}$ of Section 2. The $2M_1M_2 - M_1 - M_2$ columns of $\mathbf{D}$ are two-dimensional vectors, each corresponding to a displacement value between two neighboring sub-images. An expression for the matching quality, similar to the two-image case, can be derived in the multiple image case, by substituting $\mathbf{d}$ with $\mathbf{D}$ in (2) and extending the summation over all neighboring images. The optimal value $\mathbf{D}_{opt}$ of the displacement matrix can then be derived as follows:

$$\mathbf{D}_{opt} = \arg\min_{\mathbf{D}} E(\mathbf{D}) \tag{4}$$

The minimization in the equation above involves prohibitive computational complexity, since in this case a much larger search space is involved. Indeed, if each of the column vectors in $\mathbf{D}$ is of the form (1), $\mathbf{D}$ may assume $\left(\left(d_{1_{\max}} - d_{1_{\min}} + 1\right)\left(d_{2_{\max}} - d_{2_{\min}} + 1\right)\right)^{2M_1M_2 - M_1 - M_2}$ different values. Thus, computational complexity increases exponentially with respect to the number of sub-images. Moreover, calculation of $E(\mathbf{D})$ poses additional computational problems, since the overlap area W is now a multi-dimensional set.

In order to avoid exhaustive matching, certain constraints can be imposed on the way images are matched. Indeed, a faster method may be devised by performing simple matches only, i.e. matches between an image and one of its neighbors. The proposed method can be illustrated with the aid of a mosaicing example. In Fig. 3 (a) a mosaic of $M_1 = 2$ by $M_2 = 2$ sub-images is depicted. If one associates each image with a graph node and each local matching of two sub-images with an edge, the mosaicing of the four images can be represented by the graph of Fig. 3 (b). Computation of

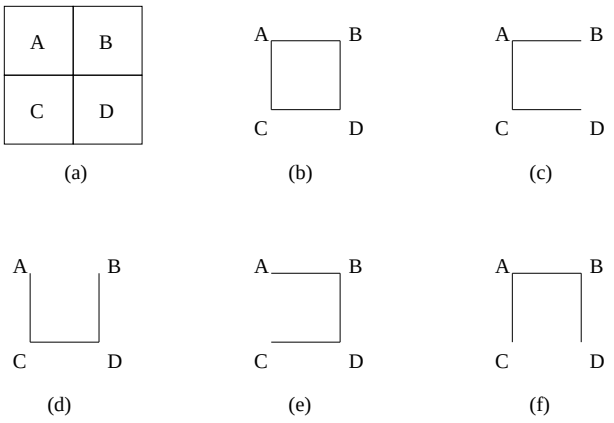

Figure 3: (a) A mosaic of $M_1 = 2$ by $M_2 = 2$ sub-images, (b) the corresponding graph, (c)-(f) the four possible spanning trees.

$\mathbf{D}_{opt}$ requires an exhaustive search in 8-dimensional space. To reduce this complexity, the entire mosaicing process is decomposed into simpler steps each involving the mosaicing of two images at a time. *Spanning trees* can provide a representation of the possible mosaicing procedures, under this constraint. Figs. 3 (c)-(f) illustrate the four different spanning trees that correspond to the graph of Fig. 3 (b). For example, in the case depicted in Fig. 3 (d) three two-image matches should be performed: image A to C, D to C, and D to B, while Figs. 3 (c), 3 (e), 3 (f) illustrate the other three possible mosaicing procedures. The final image is the one obtained by the procedure which results in the smallest matching error. Obviously, this procedure is sub-optimal but offers a significant decrease in computational complexity. The number of spanning trees that correspond to a certain graph can be calculated by the matrix-tree Theorem [12]:

Let G be a non-trivial graph with adjacency array $\mathbf{A}$ *and degree array* $\mathbf{C}$*. The number of the discrete spanning trees of G is equal with each cofactor of array* $\mathbf{C} - \mathbf{A}$.

Both $\mathbf{A}$ and $\mathbf{C}$ are matrices of size $(M_1 M_2) \times (M_1 M_2)$. If node v_i is adjacent to node v_j, $\mathbf{A}(i,j) = 1$, otherwise $\mathbf{A}(i,j) = 0$. Additionally, the degree matrix is of the form:

$$\mathbf{C} = \mathrm{diag}(d(v_1), \ldots, d(v_{M_1 M_2})) \tag{5}$$

where $d(v_i)$ denotes the number of nodes adjacent to v_i. The number of trees that correspond to graphs of sizes up to 5 by 5 images are tabulated in Table 1.

Table 1: Number of spanning trees in a graph-grid of size $M_1 \times M_2$

		M_2			
	1	**2**	**3**	**4**	**5**
1	0	1	1	1	1
2	1	4	15	56	209
M_1 **3**	1	15	192	2415	30305
4	1	56	2415	100352	4140081
5	1	209	30305	4140081	5.6×10^8

The proposed spanning tree mosaicing (STM) procedure is outlined below:

1. For each pair of neighboring images calculate the optimal displacement and the associated matching error.
2. For each spanning tree that is associated with the specific graph, calculate the corresponding total matching error, by summing the local matching errors which are associated with the two-image matches represented by the given tree.
3. Select the tree associated with the smallest total matching error.
4. Perform mosaicing of two images at a time, following a path on the selected spanning tree.

It should be clarified once again that sub-optimal results are obtained by the STM procedure, i.e., only an approximation $\hat{\mathbf{D}}_{opt}$ of the optimal matrix is computed. However, this is compensated by the significant speed gains provided by the algorithm.

As will be shown in Section 5, similar results are obtained by using either the MMAE or the MMSE criterion. Thus, MMAE may be preferred since it is faster to compute.

4. SUB-GRAPH SPANNING TREE MOSAICING

By observing Table 1 one can easily notice that, for large values of M_1 and M_2, STM becomes computationally demanding, since the number of trees grows very fast with respect to grid size. The second approach that is proposed, namely the *Sub-graph STM (SGSTM)* may, partially, address this issue. In SGSTM, a graph is partitioned into sub-graphs, by splitting the original graph vertically and/or horizontally. A sample partitioning of this type is depicted in Fig. 4. By splitting the graph vertically and then horizontally, four sub-graphs result. STM can be applied separately to each one of the four sub-graphs of Fig. 4 (d). Using the data in Table 1, it can be easily shown that a total of $192 + 1 + 0 + 1 = 194$ spanning trees should be examined. Since four images will be produced by the STM process (one for each sub-graph), a further STM step will be required, in order to mosaic these four images into the final image. Thus, 4 more trees should be added to the trees examined in the previous step, for a total of 198 trees. In contrast, an STM of the original image set would require matching error calculations for 100352 trees (see Table 1).

If the original graph is of size $M \times M$ ($M = 2^v$), the image can be gradually mosaiced by decomposing the original graph into an appropriate number of 2×2 sub-graphs, performing STM on each one, decompose once more the resulting $\frac{M}{2} \times \frac{M}{2}$ graphs and so on, until one image emerges. After mosaicing a partition's sub-graphs, new displacement matrices that correspond to the resulting sub-images should be calculated. The number of 2×2 graphs in this procedure is equal to $\frac{M}{2}\frac{M}{2} + \frac{M}{4}\frac{M}{4} + \ldots + 1 = \frac{M^2-1}{3}$. Since four spanning trees exist for a 2×2 graph, the matching error of $4\frac{M^2-1}{3}$ trees should be evaluated. Theoretical speedup values of SGSTM over STM are depicted in Fig. 5. It is obvious that SGSTM introduces large computational savings over the STM approach. Obviously, the quality of SGSTM mosaicing may be inferior to the one provided by STM, since SGSTM involves the examination of a significantly smaller number of possible sub-image matches.

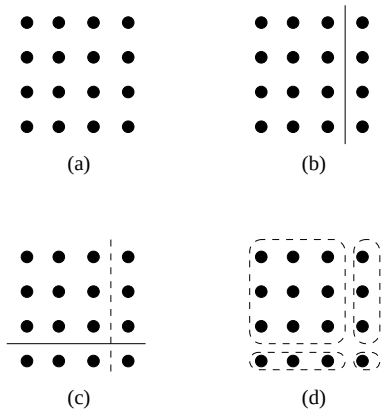

Figure 4: (a) A graph corresponding to an $M_1 = 4$ by $M_2 = 4$ image mosaic. (b) Vertical split of (a). (c) Horizontal split of (b). (d) Resulting partition of graph.

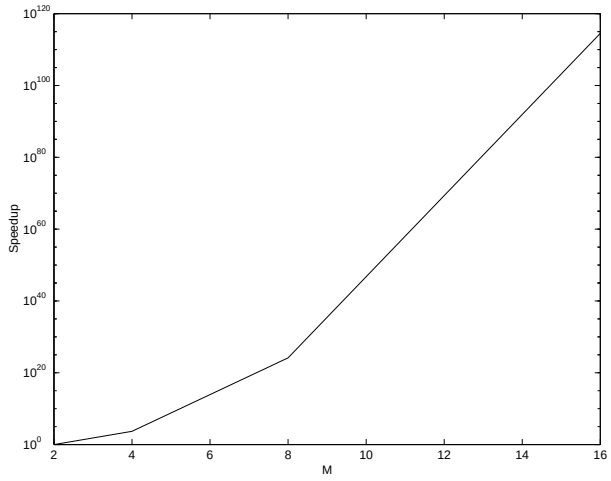

Figure 5: Theoretical speedup of SGSTM over STM, for graphs of size $M \times M$ ($M = 2^V$).

5. EXPERIMENTAL RESULTS

Simulations were carried out in order to assess the performance of the proposed methods, on several image sets. In the following, comments and results are presented for two of these sets.

The first set consisted of the 6 sub-images of Fig. 1 (a), which were arranged in $M_1 = 3$ rows of $M_2 = 2$ sub-images, each having resolution of 238×318 pixels. For each one of the $2M_1M_2 - M_1 - M_2 = 7$ pairs of neighboring sub-images, matching errors were calculated under both the MMAE and the MMSE criteria. The 2-d logarithmic search was utilized in order to obtain the optimal displacement. Subsequently, for each one of the 15 spanning trees that correspond to this 2×3 node graph, the total matching error (MMAE and MMSE) was calculated. The optimal reconstructed images under the MMAE criterion for both STM and SGSTM are depicted in Fig. 1 (b),(c).

The total time required to find all spanning trees that correspond to the given graph, calculate optimal neighboring image displacements, and output the overall matching error of each tree (for both STM and SGSTM) is tabulated in Table 2, in seconds. The corresponding matching errors are also presented in this table. In the case of SGSTM, the original 3×2 graph was partitioned into four sub-graphs. Two of them had sizes 2×1, while the last two consisted of one node each.

Table 2: STM and SGSTM performance results under the MMAE and MMSE criteria (image set 1).

	MMAE		*MMSE*	
Method	*Error*	*Time*	*Error*	*Time*
STM	27.41	2.19	1904	3.43
SGSTM	29.60	1.95	2396	2.20

As expected, the MMAE matching criterion proved to be faster in both approaches (STM and SGSTM). Speedup (i.e. the ratio of MMSE to MMAE time) was 1.13 for SGSTM and 1.57 for STM. From Table 1 it is obvious that matching error calculation is required for $(1+1+0+0)+4 = 6$ trees in the SGSTM case, instead of the 15 trees of the STM case. However the approximately 2.5 theoretical speedup factor of SGSTM over STM is not attained, because quoted time figures take into account the amount of time required to re-compute displacement matrices. Due to this overhead, speedup factors of 1.12 (when MMAE is used) and 1.56 (when MMSE is used) were recorded.

The second set consisted of 12 sub-images, which were arranged on a grid of $M_1 = 4$ rows of $M_2 = 3$ sub-images each. Each sub-image had a resolution of 951×951 pixels. For this graph 2415 spanning trees exist. Similar to the previous set, for each one of the $2M_1M_2 - M_1 - M_2 = 17$ pairs of neighboring sub-images, matching errors were calculated under both the MMAE and the MMSE criteria and the 2-d logarithmic search was utilized in order to obtain the optimal displacement. Subsequently, for each one of the 2415 spanning trees, the total matching error (MMAE and MMSE) was calculated.

The total time that was required to find all spanning trees, calculate optimal neighboring image displacements, and output the overall matching error of each tree in this case is presented in Table 3. In the case of SGSTM, the graph was decomposed into four sub-graphs: two of size 2×2 and two of size 2×1, which required the calculation of the total error for 14 trees, compared to the 2415 of STM. It is evident that, in this image set, SGSTM is more than an order of magnitude faster than STM. More specifically, the speedup provided by the SGSTM method over the STM method was 13.9 when the MMSE criterion was used and 11.4 for the MMAE criterion. Of course, this speedup was accompanied by a significant increase in the matching error. Furthermore, MMAE proved once again to be computationally more efficient than MMSE. Sample minimum, maximum, mean and variance of matching error, over the entire spanning tree set for the STM approach, are recorded in Table 4.

The twenty spanning trees that exhibited the lowest MMAE scores for this image set are depicted in Fig. 6. By studying these spanning trees as well as the MMSE and MMAE scores, the following observations can be made:

- MMAE optimality is closely related to MMSE optimal-

Table 3: STM and SGSTM performance results under the MMAE and MMSE criteria (image set 2).

	MMAE		*MMSE*	
Method	*Error*	*Time*	*Error*	*Time*
STM	6.4	782.7	107	1265
SGSTM	9.7	68.9	228	91

Table 4: STM matching error statistical measures (image set 2).

Measure	*MMAE*	*MMSE*
Maximum	10.7	378
Minimum	6.4	107
Mean	7.9	186
Variance	0.45	1771

ity. Indeed, the trees that exhibited the lowest MMAE figures, exhibited also the lowest MMSE figures.

- The most characteristic feature of the trees with the lowest error figures was that optimal matching began from the center and proceeded outwards. In other words, in these trees, the central nodes of the graph were connected. On possible explanation is that the matching quality of the central nodes is more crucial to the overall mosaicing quality, than the matching quality of the other nodes. This issue is currently under investigation.

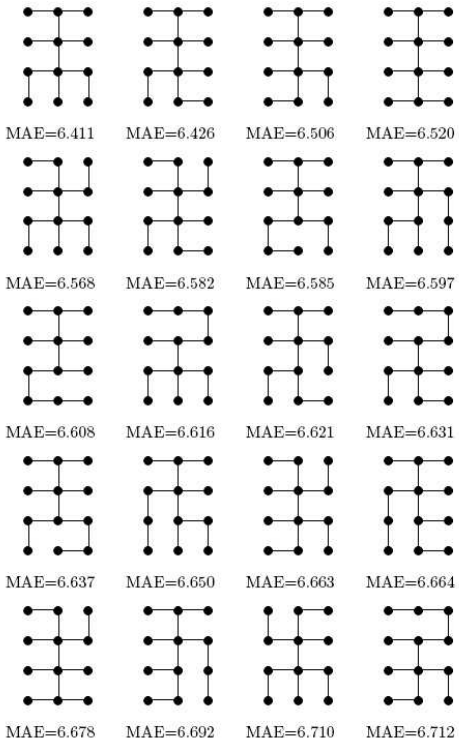

Figure 6: The twenty spanning trees that produced the lowest MMAE figures.

6. CONCLUSIONS

Two novel methods that can be used for the computationally efficient mosaicing of sets of sub-images (e.g. in applications involving high resolution digitization of works of art) were proposed in this paper. The proposed methods utilize spanning trees in order to describe the order of the mosaicing process. Exhaustive search for the optimal placement of sub-images with respect to each other is avoided by examining only matches between pairs of neighboring images. The SGSTM method is significantly faster than the STM method but this speedup comes at the cost of increased matching error. However, SGSTM can be utilized for fast visualization of mosaicing results (e.g. mosaic previews).

REFERENCES

[1] M. Corsini, F. Bartolini and V. Cappellini, "Mosaicing for high resolution acquisition of paintings", in *Proc. 7th International Conference on Virtual Systems and Multimedia*, pp. 39 - 48, 2001.

[2] W. Puech, A. G. Bors, I. Pitas and J-M. Chassery, "Projection distortion analysis for flattened image mosaicing from straight uniform generalized cylinders", *Pattern Recognition*, vol. 34, no. 8, pp. 1657-1670, August 2001.

[3] V. Swarnakar, M. Jeong, R. Wasserman, E. Andres and D Wobschall, "Integrated distortion correction and reconstruction technique for digital mosaic mammography", in *Proc. SPIE Medical Imaging 1997: Image Display*, Vol. 3031, p. 673-681, 1997.

[4] H.-Y. Shum and R. Szelisky", Construction of Panoramic Image Mosaics with Global and Local Alignment", *International Journal of Computer Vision*, vol 36, no 2, pp. 101-130, February 2000.

[5] H. S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion, in *Proc. CVPR 1997*, pp. 450-456, 1997.

[6] S. Peleg and J. Herman, " Panoramic Mosaics with VideoBrush", in *Proc IUW-97*, pp. 261-264, May 1997.

[7] F. Toyama, K. Shoji and J. Miyamichi, "Image mosaicing from a set of images without configuration information", in *Proc. ICPR 2004*, vol 2, pp. 899 - 902, 2004.

[8] Y. Gehua and C.V. Stewart, "Covariance-driven mosaic formation from sparsely-overlapping image sets with application to retinal image mosaicing", in *Proc. CVPR 2004*, vol. 1, pp. 804-810, 2004.

[9] Yong Li, Daiyin Zhu, Zhaoda Zhu and Ling Wang, "Automatic mosaicing for airborne SAR imaging based on subaperture processing", in *Proc. IGARSS '05*, vol. 7, pp. 4644 - 4647, 2005.

[10] O. Pizarro and H. Singh, "Toward large-area mosaicing for underwater scientific applications", *IEEE Journal of Oceanic Engineering*, vol. 28, no. 4, pp 651-672, October 2003.

[11] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press, 1988.

[12] N. L. Biggs, E. K. Lloyd and R. J. Wilson, *Graph Theory 1736-1936*, Clarendon Press, 1986.