Image Interpolation by Joint View Triangulation

Maxime LHUILLIER Long QUAN
CNRS-GRAVIR-INRIA
ZIRST – 655 avenue de l'Europe
38330 Montbonnot, France

Abstract

Creating novel views by interpolating prestored images or view morphing has many applications in visual simulation. We present in this paper a new method of automatically interpolating two images which tackles two most difficult problems of morphing due to the lack of depth information: pixel matching and visibility handling. We first describe a quasi-dense matching algorithm based on region growing with the best first strategy for match propagation. Then, we describe a robust construction of matched planar patches using local geometric constraints encoded by a homography. After that we introduce a novel representation—joint view triangulation—for visible and half-occluded patches in two images to handle their visibility during the creation of new view. Finally we demonstrate these techniques on real image pairs.

1 Introduction

There has been increased interest both for computer vision and computer graphics in image-based rendering methods, which deal with how to produce an image from an arbitrary new point of view, given a set of reference images. Compared with the classical rendering methods [27] based on geometry and developed in computer graphics, the images produced by image-based rendering systems are more photorealistic, and the production is almost real-time. We will develop in this paper a new method of generating in-between views from two reference images. This kind of view interpolation techniques is also called morphing [28].

Reconstruction-based methods: Classical 3D reconstruction techniques developed in computer vision is a natural solution to image-based rendering by transferring image textures onto it. Synthesizing arbitrary views of the scene consists of only reprojecting the rendered 3D model for any given point of view. More recent methods without explicit 3D reconstruction have shown that using matching constraints of multiple images—fundamental matrix for two-view and trifocal tensor for three-view, the pixel location on the synthesized view could be predicted from the pixel locations on the reference images. Laveau and Faugeras [12] used the fundamental matrices and Avidan and Shashua [1] used the trifocal tensor. These methods using matching tensors are still essentially equivalent to reconstruction methods as they are nothing but implicit 3D reconstruction methods. All these methods having focused on rigid scenes could be considered as reconstruction-based methods, either explicitly or implicitly.

Interpolation-based methods: Primarily in computer graphics, view interpolation or morphing techniques as a mean to generate smooth transitions between reference images by simply interpolating each pixel color from the first to the second image value have been developed. In the field morphing of Beier and Neely [2], line segments specified and matched by an animator are used to create interpolated images. Lee et al. [13] studied different warping strategies for morphing. Chen et al. [4, 3] popularise, later through QuickTimeVR products, the idea of direct pixel by pixel interpolation, however they originally assume that the pixel correspondences in the basis images are given since the basis images are computer rendered. Seitz and Dyer [20] investigated view interpolation but are concerned with physically-valid view generation via rectification of perspective image pair following the linear combination method developed for object recognition of affine images. It is also aimed at rendering rigid scene, therefore related to the reconstruction-based method. More general and time/memory consuming methods based on sampling plenoptic functions have also been developed [14, 8] in computer graphics community.

The in-between images produced by morphing techniques look strikingly life-like especially for face images [2, 21], while reconstruction-based methods have to struggle with the problem of occlusion. However, morphing techniques only produce the images of restricted points of view compared with theoretically arbitrary point of view that reconstruction-based methods could. Morphing techniques except that of Seitz and Dyer [21] generally do not guarantee physical validity of the resulting new image, but they could equally handle deformable objects that reconstruction based methods could not.

The principal weakness of the actual morphing techniques due to mainly the lack of depth information for photographic images is that the hardest correspondence problem is avoided by either using human animator [2] or using computer generated range data [4]. This motivates us to describe a new morphing method in this paper. The first step is to develop an automatic matching algorithm for sufficiently textured images. The matching method starts from the most reliable matches and propagates the best match in its neighboring pixels in a similar way to region growing technique [17] used for image segmentation [9]. This gives us a what we call quasi-dense matching map. Then applying the piecewise smooth assumption, we construct robustly matched planar patches across two images. After that, a joint view triangulation is defined and a robust algorithm for computing it is proposed to separate matched areas from unmatched ones and handle the partially occluded areas. The joint view triangulation is inspired by impostors [23] and mesh integration for range data [26, 24] from computer graphics community, but it is completely different, since we do not use any 3D input data that is easily available for the computer graphics work. More details could be found in the technical report [16].

2 Quasi-dense matching

Establishing correspondence between two images either for high level image primitives such as points and line segments or for dense pixel-by-pixel matching is probably the most challenging problem in computer vision. Many work has been devoted for this task [11, 5], in particular for stereo images in which the search space reduces to 1D along the epipolar lines. Many potential vision applications are moderated due to the lack of reliable matching algorithms. Any general purpose and reliable matching algorithm is still far from realistic. Meanwhile, it is well known from many years' accumulated experience, that no matter which matching algorithms we choose, we could expect reliable results solely on textured areas of the image. This is not surprising since there is just no enough information available in the non textured areas to make the correct decision. This motivates us the definition of a quasi-dense matching which is meant that the matching map between images could never be truly dense, it could only consists of a set of sparsely distributed dense regions. The quasi-dense matching such defined is therefore a more realistic goal, thus could be more reliably computed than the full dense matching.

In the remaining of this section, we will develop a quasi-dense matching algorithm for a pair of images. We do not use any strong geometric constraints, therefore no rigidity assumption of the scene is required. We accepte also larger image distance unlike the case of small base-line of stereo images.

The method starts from matching some points of interest which have the highest textureness as seed points to bootstrap a region growing type algorithm to propagate the matches in its neighborhood from the most textured (therefore most reliable) pixels to less textured ones. The algorithm could therefore be described in two steps: Seed selection and propagation.

- Seed selection

  Points of interest [22, 15, 10] are naturally good seed point candidates, as points of interest are by its very definition image points which have the highest textureness.

  So we first extract points of interest from two original images, then we use a ZNCC correlation method to match the points of interest across two images. The $ZNCC_{\mathbf{x}}(\Delta)$ (zero-mean normalized cross-correlation) at each point $\mathbf{x} = (x, y)^T$ with the shift $\Delta = (\Delta_x, \Delta_y)^T$ is defined to be

  $$\frac{\sum_{\mathbf{i}}(I(\mathbf{x}+\mathbf{i}) - \bar{I}(\mathbf{x}))(I'(\mathbf{x}+\Delta+\mathbf{i}) - \bar{I}'(\mathbf{x}+\Delta))}{(\sum_{\mathbf{i}}(I(\mathbf{x}+\mathbf{i}) - \bar{I}(\mathbf{x}))^2 \sum_{\mathbf{i}}(I'(\mathbf{x}+\Delta+\mathbf{i}) - \bar{I}'(\mathbf{x}+\Delta))^2)^{1/2}}$$

  where $\bar{I}(\mathbf{x})$ and $\bar{I}'(\mathbf{x})$ are the means of pixel values for the given window centered at $\mathbf{x}$. This gives us the initial list S of correspondences sorted by the correlation score.

- Propagation

  Let M be the list of the current matched points, B be the list of current seeds. Obviously, The list B is initialized to S and the list M to the empty list.

  At each step, we pull the best match $m \leftrightarrow m'$ from the set of seed matches B. Then we look for additional matches in the neighborhood of m and m'. The neighbors of m is taken to be all pixels within the 5×5 window centered at m. For each neighboring pixel of the first image, we first construct in the second image a list of tentative match candidates which consists of all pixels of a 3×3 window in the neighborhood of its corresponding location in the second image (see Figure 1). The matching criterion $c(\mathbf{x}, \mathbf{x}')$ is still the correlation defined above but within a 5×5 window.

  Finally additional matches in the neighborhood of m and m' are added simultaneously in the match list M and the seed match list B such that the unicity constraint is preserved. The algorithm terminates when the seed match list B becomes empty.

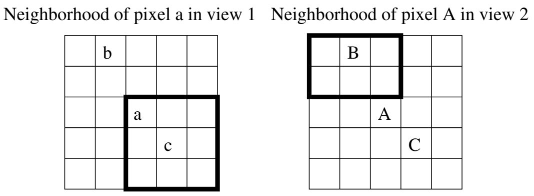

Figure 1. The neighborhood is limited to the 5×5 window around the pixels a and A. The tentative matches for b (resp. C) are all pixels inside the black frame in the second (resp. the first) image.

This algorithm could be efficiently implemented with a heap data structure for the seed pixels B of the regions of the matched points. Notice that as each time only the best match is selected, this drastically limits the possibility of bad matches. For instance, the seed selection step seems very similar to many existing methods [29, 25] for matching points of interest using correlation, but the crucial difference is that we need only to take the most reliable ones rather than trying to match a maximum of them. In some extreme cases, only one good match of points of interest is sufficient to provoke an avalanche of the whole textured images. This makes our algorithm much less vulnerable. The same is true for propagation, the risk of bad propagation is considerable diminished by the best first strategy over all matched boundary points.

3 Robust construction of planar patches using local geometric constraints

The brut quasi-dense matching result may still be corrupted and irregular. Though we do not have any rigidity constraint on the scenes, we assume that the scene surface is at least piecewise smooth. Therefore, instead of using global geometric constraints encoded by fundamental matrix or trifocal tensor, we could use local geometric constraints encoded by planar homography. The quasi-dense matching is thus regularised by locally fitting planar patches. The construction of the matched planar patches is described as follows.

The first image is initially subdivided into square patches by a regular grid of two different scales 8×8 and 16×16.

For each square patch, we obtain all matched points of the square from the quasi-dense matching map. A plane homography $\mathbf{H}$ is tentatively fitted to these matched points $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$ of the square to look for potential planar patches. A homography in $\mathcal{P}^2$ is a projective transformation between projective planes, it is represented by a homogeneous 3×3 non singular matrix such that $\lambda_i \mathbf{u}'_i = \mathbf{H}\mathbf{u}_i$, where $\mathbf{u}$ and $\mathbf{u}'$ are represented in homogeneous coordinates. Each pair of matched points provides 2 homogeneous linear equations in the matrix entries h_{ij}. The 9 entries of the homography matrix counts only for 8 d.o.f. up to a scale, therefore 4 matched points, no three of them collinear, are sufficient to estimate the $\mathbf{H}$. Because a textured patch is rarely a perfect planar facet except for manufactured objects, the putative homography for a patch can not be estimated by standard least squares estimators. Robust methods have to be adopted, which provide a reliable estimate of the homography even if some of the matched points of the square patch are not actually lying on the common plane on which the majority lies. The Random Sample Consensus (RANSAC) method originally introduced by Fischler and Bolles [6] is used for robust estimation of the homography. RANSAC has been successfully used for robust computation of the geometric matching tensors in [25].

If the consensus for the homography reaches 75%, the square patch is considered as planar. The delimitation of the corresponding planar patch in the second image is defined by mapping the four corners of the square patch in the first image with the estimated homography $\mathbf{H}$. Thus, a pair of corresponding planar patches in two images is obtained.

This process of fitting the square patch to a homography is first repeated for all square patches of the first image from the larger to the smaller scale, it turns out all matched planar patches at the end. Notice that the planar patches so constructed may overlap in the second image. To reduce the number of the overlapped planar patches, but not solve the problem, the corners of the adjacent planar patches are forced to coincide in a common one if they are close enough. This is illustrated in Figure 2.

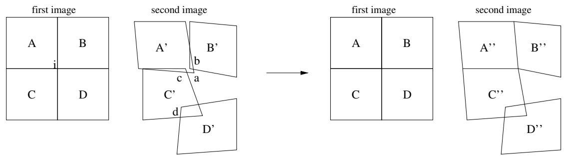

Figure 2. The patches A', B', C' and D' recomputed by the homographies in the second image correspond to the regular patches A, B, C and D in the first image. Because the corners a, b and c of different patches are very close, they are made to coincide in one common point. Note that this only improves not solve the overlapping problem of the patches, e.g. the patch C' and D' remain overlapped after this procedure.

Each planar patch could be subdivided along one of its diagonals into 2 triangles for further processing.

From now on, the meaning of a matched patch is more exactly a matched *planar* patch, as we will only consider the matched patch which succeeds in fitting a homography.

4 Joint view triangulation

Because image interpolation relies exclusively on image content with no depth information, it is sensitive to changes in visibility. In this section, we propose a multiple view representation to handle the visibility issue that we call *joint view triangulation* which triangulates simultaneously and consistently (the *consistency* will soon be precised) two images without any 3D input data. Triangulation has proven to be a powerful tool of efficiently representing and restructuring individual image or range data. However, to our knowledge, no one has yet tried to deal with the similar representation for multiple views.

The triangulation in each image will be Delaunay because of its minimal roughness properties [19]. The Delaunay triangulation will be necessarily constrained as we want to separate the matched regions from the unmatched ones. The boundaries of the connected components of the matched planar patches of the image must appear in both images, therefore are the constraints for each Delaunay triangulation.

By consistency for the joint triangulation, it is meant that there is a one-to-one correspondence between the image vertices and a one-to-one correspondence between the constrained edges—boundaries of the matched regions.

Recall that a constrained Delaunay triangulation [18] is a Delaunay triangulation in which the circumcircle of each triangle does not contain in its interior any other *'visible points'*. Two points are said to be visible if they are not separated by a constraint edge.

In summary, the joint view triangulation for two views has the following properties: 1. one-to-one vertex correspondence in two images; 2. one-to-one constraint edge correspondence in two images, the constraint edges are the boundary edge of the connected components of the matched regions in two images; 3. The triangulation in each image is a constrained Delaunay by the constraint edges.

A greedy method for joint view triangulation is a natural choice. The algorithm can be briefly described as follows.

- The joint view triangulation starts from two triangles in each image as illustrated in Figure 4.
- Then, each matched planar triangle is incrementally inserted into each triangulation. The insertion is carried out in order, row by row from the top to the bottom of the grid. For each row, a two-pass algorithm is used for implementation ease and robustness.
  - The first pass consists of examining all planar patches from left to right. If the triangle in the second image does not intersect any current matched areas, its vertices are inserted into image plane for constrained triangulation. Notice that the triangle connected to the matched area only through a common vertex is considered as intersecting due to the topology of actual data structure illustrated in Figure 3. Next, the polygonal boundary of each matched area is recomputed if the newly added triangle is connected to one of the matched areas. A triangle is connected to a matched area delineated by a polygon if it shares a common edge with the boundary polygon.

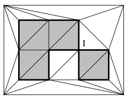

Figure 3. The gray triangles are the matched planar patches and the white ones are not matched. This configuration in which more than two constraint edges (black edges) share a vertex I is considered as overlapping to simplify the topology of the data structure.

  - A second pass for the current row is necessary to fill in undesirable unmatched holes that may be created during the first pass due to the topological limitation of the data structure mentioned above.
- completion step. up to this point, a consistent joint view triangulation is obtained. We improve the structure by further checking if each unmatched triangle could be fitted to an affine transformation. If an unmatched triangle succeeds in fitting an affine transformation, it is changed from unmatched into matched one in the joint view triangulation.

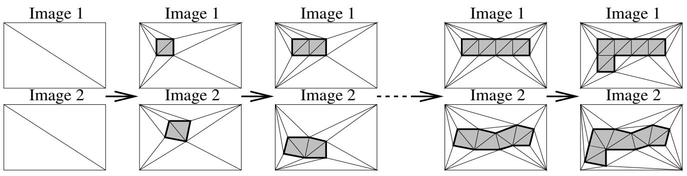

Figure 4. Illustration of the incremental construction of the joint view triangulation in the two images.

5 View interpolation

Now we describe how to generate all in-between images by interpolating the two original images. Any in-between image $I(\lambda)$ is parameterized by $\lambda \in [0, 1]$ and obtained by shape interpolation and texture bleeding of the two original images such that the two original images are the endpoints of the interpolation path, $I(0) = I$ and $I(1) = I'$.

A three-step algorithm is given as follows:

- warp individual triangle

The position is first interpolated for each vertex of the triangles $\mathbf{u} \leftrightarrow \mathbf{u}'$ as

$$\mathbf{u}''(\lambda) = (1-\lambda)\mathbf{u} + \lambda\mathbf{u}'.$$

and a weight w is assigned to each warped triangle to measure the deformation of the warped triangle. The weight w is proportional to the ratio γ of the triangle surface in the first image w.r.t. the second image bounded by 1, that is $w = min(1, \gamma)$ for the triangles of the first image and $w' = min(1, 1/\gamma)$ for the triangles of the second image.

- warp the whole image

  To correctly handle the occlusion problem of patches, we could use either Z-buffer algorithm or the Painter's method in which pixels are sorted in back to front order when the depth information was available. In the absence of any depth information, a warping order for each patch is deduced from its maximum disparity to expect that any pixels that map to the same location in the generated image are arriving in back to front order as in the Painter's method [7].

  All triangular patches of the original images I and I' are warped onto $\tilde{I}$ and $\tilde{I}'$ by first warping unmatched ones followed by matched one. The triangles whose vertices are image corners are not considered.

  At first, all unmatched triangles are warped onto $\tilde{I}$ and $\tilde{I}'$ as they include either holes caused by occlusion in the original images. More precisely, small unmatched triangles connecting matched and unmatched regions are warped before the others unmatched triangles, since they are most probably from different objects.

  Secondly, matched triangles are warped by a heuristic order which is the decreasing order of the maximum displacement of the triangle.

- Color interpolation

  The final pixel color is obtained by bleeding two weighted warped images $\tilde{I}$ and $\tilde{I}'$:

$$I''(\mathbf{u}) = \frac{(1-\lambda)w(\mathbf{u})\tilde{I}(\mathbf{u}) + \lambda w'(\mathbf{u})\tilde{I}'(\mathbf{u})}{(1-\lambda)w(\mathbf{u}) + \lambda w'(\mathbf{u})}.$$

6 Experimental results

The new method described in this paper has been demonstrated on many real image pairs. Mpeg sequences of interpolated images could be played at our Web site (`//www.inrialpes.fr/movi/pub/Demos/`). Here we look at two examples of the morphing results. One pair of images is from the public domaine *flower garden sequence*, and another is outdoor house scene taken from the mountains.

The flower garden images are chosen for two reasons: it has a big occluding front tree and it has large disparity. Figure 5 shows the original image pair with the matched points of interest. Starting from the sorted matched points of interest, the matching is propagated into the whole image.

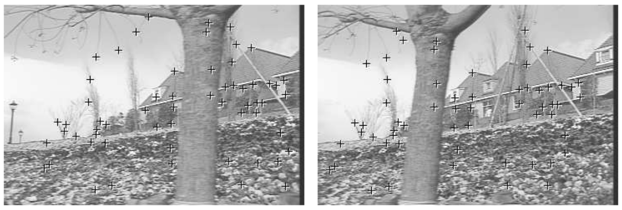

Figure 5. The two original images from the garden flower superimposed with the matched points of interest marked as crosses.

The quasi-dense matching result is shown in Figure 6 in which a gray-black checker-board is superimposed onto the original image to facilitate check matches. It can also be interpreted as the regular grid in the first image and the corresponding transformed grid in the second image. We can see the large unmatched areas such as the non-textured sky and the occluded garden and house by the front tree. These areas are connected to their matched neighbors thanks to the Delaunay triangulation.

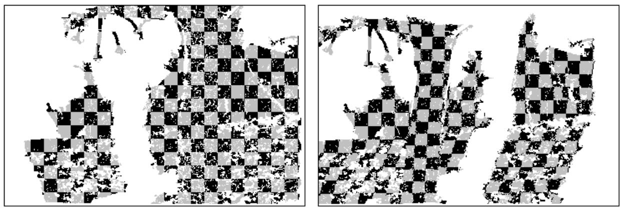

Figure 6. The result of quasi-dense matching: a regular gray-black chess-board is drawn over the first image, the deformed chess-board in the second image helps to illustrate the corresponding points.

The joint view triangulation result is illustrated in Figure 7. The black edges in the figure delimit the matched areas and they are also the constraints for Delaunay triangulation. The white edges are Delaunay and are not necessarily matched.

The whole algorithm runs for the garden flower images (360×240) within 8.4 seconds including 2.6 for seeds, 3 for propagation and 2.8 for the joint view triangulation on a UltraSparc 300MHz.

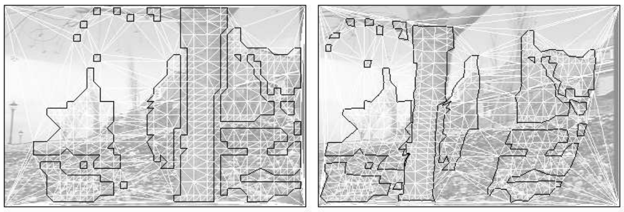

Figure 7. The result of the joint view triangulation: the black edges are the boundary edges of the matched regions and the white edges are Delaunay edges which are not matched.

Some in-between generated images are shown in Figure 8. If we examine carefully the generated images, we can see the precision at the occlusion borders is bounded by the patch size, roughly 8 pixels. There are also some unmatched triangles appeared inside the matched areas due to incorrect dense matching.

The house image pair shown in Figure 9 is a typical textured outdoor scene. It is very different from the garden flower sequence in that the matching is much harder due to the fine texture of the grasses, but the interpolation is easier thanks to that the ordering constraint of the matches is preserved.

Figure 9. The image pair of the house image sequence superimposed with the matched points of interest marked as crosses.

The quasi-dense matching result for the house images is shown in Figure 10. We can notice that there are some wrong matches around the left corner of the green grass, also few sparse wrong ones occur in the front bushes.

Figure 11 shows the joint view triangulation with the matched planar patches. Particularly in the grass and bushes cases, the main goal of the joint triangulation is the interpolation of enclosed unmatched areas.

Some sample images of the synthesized sequence are shown in Figure 12.

The whole algorithm runs for the house images (768×512) within 57.6 seconds including 19 for the initial matches, 20.6 for the quasi-dense matching and 18 for the joint view triangulation on a UltraSparc 300MHz.

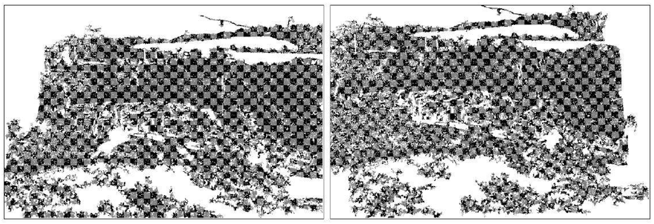

Figure 10. The result of quasi-dense matching: a regular gray-black chess-board is drawn over the first image, the deformed chess-board in the second image helps to illustrate the corresponding points.

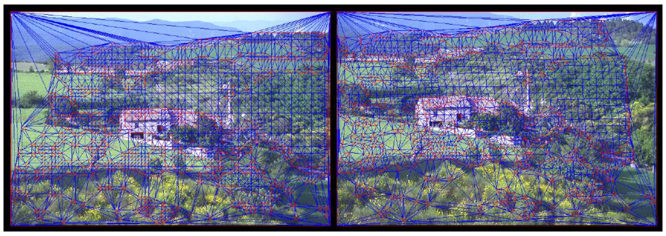

Figure 11. The result of the joint view triangulation: the red edges are the boundary edges of the matched regions and the blue ones are Delaunay edges which are not matched.

7 Conclusion and future work

We have presented a new method of automatic image interpolation in which we contributed to the two most difficult problems for the automatic image interpolation: correspondence and visibility representation. We proposed a quasi-dense matching based on region growing followed by a regularisation procedure using local constraints encoded by homography for correspondence problem. A joint view triangulation was introduced to handle the patch visibility during the new view generation step. These techniques have produced visually very convincing sequences on real image pairs.

There are still many topics related to morphing and reconstruction which we plan to improve and investigate. For instance, a refinement of the boundaries of the matched areas is necessary as its current accuracy is limited by the patch size. A variable patch size could be a good solution to distinguish large untextured areas from manufactured object parts and the thin objects. Quasi-dense matching could be further completed by matching small non-textured areas after joint view triangulation. Finally the extension to N views is actually under investigation.

Figure 8. Some sample images of the interpolation: $\lambda = 0, 0.25, 0.5, 0.75$ and 1 from the left to the right.

Figure 12. Some sample images of the interpolation: $\lambda = 0, 0.25, 0.5, 0.75$ and 1 from the left to the right.

Acknowledgements

We would like to thank Roger Mohr for many fruitful discussions.

References

[1] S. Avidan and A. Shashua. Novel view synthesis in tensor space. CVPR'97.

[2] T. Beier and S. Neely. Feature-based image metamorphosis. *SIGGRAPH'92*.

[3] S.E. Chen. Quicktime VR - an image-based approach to virtual environment navigation. SIGGRAPH'95.

[4] S.E. Chen and L. Williams. View interpolation for image synthesis. *SIGGRAPH'93*.

[5] U.R. Dhond and J.K. Aggarwal. Structure from stereo – a review. IEEE TSMC,19(6),1989.

[6] M.A. Fischler and R.C. Bolles. RANSAC. *GIP*, 24(6), 1981.

[7] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics : Principles and Practice*.

[8] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. SIGGRAPH*'96*.

[9] R.M. Haralick and L.G. Shapiro. Image segmentation techniques. *CVGIP*, 1985.

[10] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*'88.

[11] A. Koschan. What is new in computational stereo since 1989 : A survey on stereo papers. TR93-22 University of Berlin, 1993.

[12] S. Laveau and O. Faugeras. 3D scene representation as a collection of images and fundamental matrices. TR2205, INRIA 1994.

[13] S.Y. Lee, K.Y. Chwa, J. Hahn, and S.Y. Shin. Image morphing using deformation techniques. *JVCA*, 1996.

[14] M. Levoy and P. Hanrahan. Light field rendering. SIGGRAPH*'96*.

[15] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*'81.

[16] M. Lhuillier. Towards automatic interpolation for real and distant image pairs. Inria RR-3619. 1999. (`//www.inria.fr/RRRT/RR-3619.html`).

[17] G.P. Otto and T.K. Chau. A region-growing algorithm for matching of terrain images. *IVC*, 7(2), 1989.

[18] F. Preparata and M.I. Shamos. *Computational Geometry, An Introduction*. 1985.

[19] D. Rippa. Minimal roughness property of the delaunay triangulation. *CAGD*, 7, 1990.

[20] S.M. Seitz and C.R. Dyer. Physically-valid view synthesis by image interpolation. *Workshop on RVS*, 1995.

[21] S.M. Seitz and C.R. Dyer. View morphing. SIGGRAPH*'96*.

[22] J. Shi and C. Tomasi. Good features to track. *CVPR*'94.

[23] F. Sillion et al. Efficient impostor manipulation for real-time visualization of urban scenery. *Eurographics'97*.

[24] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *PAMI*, 17(4):344–358, 1995.

[25] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. BMVC'96.

[26] G. Turk and M. Levoy. Zippered polygon meshes from range images. In SIGGRAPH94, pages 311–318, 1994.

[27] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques*. 1992.

[28] G. Wolberg. *Digital Image Warping*. 1990.

[29] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. TR2273 INRIA, 1994.