Mult-View Image Compositions

Lihi Zelnik-Manor and Pietro Perona
California Institute of Technology
Pasadena, CA, 91125, USA
{lihi,perona}@vision.caltech.edu

March 22, 2007

Abstract

The geometry of single-viewpoint panoramas is well understood: multiple pictures taken from the same viewpoint may be stitched together into a consistent panorama mosaic. By contrast, when the point of view changes or when the scene changes (e.g., due to objects moving) no consistent mosaic may be obtained, unless the structure of the scene is very special.

Artists have explored this problem and demonstrated that geometrical consistency is not the only criterion for success: incorporating multiple view points in space and time into the same panorama may produce compelling and informative pictures. We explore this avenue and suggest an approach to automating the construction of mosaics from images taken from multiple view points into a single panorama. Rather than looking at 3D scene consistency we look at image consistency. Our approach is based on optimizing a cost function that keeps into account image-to-image consistency which is measured on point-features and along picture boundaries. The optimization explicitly considers occlusion between pictures.

We illustrate our ideas with a number of experiments on collections of images of objects and outdoor scenes.

1 Introduction

A single picture cannot always capture the full scene. It has thus become common among artists and amateur photographers to take multiple pictures of the same scene and compose them into mosaics. When all the pictures are taken from a single view point the geometry of the panorama is well understood [6, 12, 14]. This, together with methods for matching informative image features [8] and good blending techniques [3, 4] have made it possible for any amateur photographer to produce automatically mosaics of photographs covering very wide fields of view [2, 12]. By contrast, when the point of view changes or when objects moved in the scene, no consistent mosaic may be obtained, unless the structure of the scene is very special.

Artists have explored this problem and demonstrated that incorporating multiple view points into the same mosaic may produce more informative representations than a single view point panorama can. For example, see Figure 6.a. This fresco by Paolo Uccello shows the podium as if the viewer is looking upward to it, yet the rider and horse are painted from a direct side view. While the painter has the freedom to change the view point smoothly, this is not always possible when stitching pictures. One cannot expect to always get the smooth appearance of a single view panorama when mosaicing pictures with view point changes. Nevertheless, artists like David Hockney and James Balog have demonstrated that mosaics with visible inconsistencies across picture boundaries can nevertheless look compelling and informative. Such mosaics have become common also within amateur photographers (for example go to http://www.flickr.com/ and search for pictures tagged with "composites", "hockney" or "joiner"). Figure 1 shows some examples.

There are scenarios in which multi view point mosaics must be used because there is no other option. For example, often one cannot capture the full scene from a single

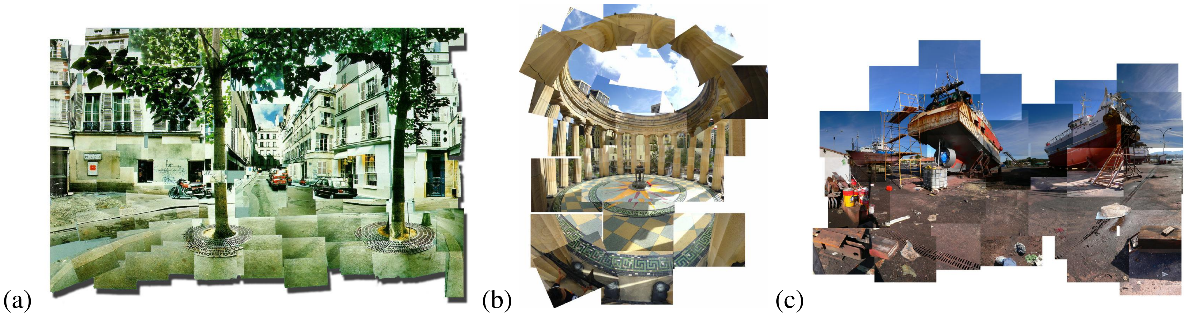


Figure 1: **Multi view art work:** (a) David Hockney's "Place Furstenberg", Paris, 1985. (b),(c) Two sample compositions downloaded from flickr.com, constructed by amateur photographers. Many more can be found on the web.

view point due to occlusions, see Figure 7.a. Changes in view point can also result from people moving while being photographed, see Figures 8.a,c. We thus believe that automatic construction of mosaics from pictures with view points changes is a required tool.

Mosaics incorporating multiple view points have been explored before. Wood et al. [13] suggested an approach to computerized design of multiperspective panoramas for cel animation where all viewpoints are available apriori. A wide range of approaches have been suggested for constructing multi view panoramas when the input is a video sequence taken by a smoothly moving video camera, e.g., [11, 10, 15, 7].

In this paper, however, our goal is to create compositions of discrete sets of photographs taken from highly different view points. Rather than assuming a strong model on the geometry of picture taking (e.g. single view point or perhaps 3D reconstruction of viewpoints and shape [5]) we look for consistency in composition space. Inspired by artists, we suggest an approach that aligns and stacks (orders the layers of) pictures by minimizing visible artifacts in the final composition. Thus we do not assume an underlying '3D reality' but rather take the point of view that the final picture is a composition which is driven by the photographs.

The rest of the paper is organized as follows. We start by outlining the overall framework in Section 2. We then proceed and describe in detail the various steps of the approach in Sections 3,4,5,6. We conclude in Section 7. Our ideas are illustrated through experiments which appear throughout the text.

2 Overall Framework

When pictures are taken from different view points there is no globally consistent way to obtain a composition, therefore we cannot hope to obtain geometrical consistent matches between neighboring pictures. We replace the geometrical distortion cost of Brown and Lowe [2] with a cost that is a combination of geometrical and appearance consistency. Furthermore, we measure geometrical and appearance consistency directly on the composition plane, rather than on the viewing sphere. We require appearance consistency because sometimes pictures that are geometrically inconsistent may easily blend into each other, e.g., when there is texture or uniform color near the picture boundary. Alignment errors in this case should thus be penalized less than when the error is very salient.

We furthermore notice that geometrical and appearance inconsistencies that are hidden from view have little importance, as compared to those that are visible. Our optimization takes this into account.

The suggested framework consists of the following steps:

1. For each pair of images find point-feature correspondences and fit a similarity transformation between them. Keep only correspondences which can be approximately aligned by the transformation (Section 3).

2. Find global alignment of the images in the composition by minimizing distances between correspondences. If importance weights were assigned to the

correspondences, incorporate them in the optimization process (Section 3).

3. Find the best layering of the images: search over all possible orders the one which minimizes discontinuities across image boundaries in the composition (Section 4).

4. Assign high weights to correspondences near visible image boundaries and low weights otherwise (Section 5).

5. Repeat steps 2 to 4 until weights and transformations are not updated.

6. If desired, blend images only near visible seams (Section 6).

In the following sections we describe in detail each of the above steps.

3 Image Alignment

For image alignment we adopt the feature-based technique suggested by Brown & Lowe [2], with one major difference. In [2] images were assumed to be taken from a single viewpoint, thus alignment was obtained on the viewing sphere by solving for the camera rotation at each image. This approach is inadequate for images taken from multiple view points. Instead we optimize the alignment on the 2D panorama canvas by solving for a similarity transformation for each image. That is, we allow images to translate, scale and rotate.

The choice of similarities is motivated by the beautiful compositions we have found on the web (e.g., Figure 1), as well as by our own experience. We have collected tens of image datasets and composited them manually limiting the transformations to translation, scale and rotation, which are the basic available tools in most image editing software. We have found this set of transformations to be sufficient and thus adopt it in our automatic framework.

Following Brown & Lowe, we first extract and match SIFT features [8] between all pairs of images. We then use RANSAC [6] to select a set of inliers that are compatible with a similarity transformation between each pair of images. Next we apply the probabilistic model suggested in [2] to verify the match. We discard all feature matches which are not geometrically consistent with the transformation between the images (RANSAC outliers). Finally, given the set of geometrically consistent matches between the images, we use bundle adjustment [2] to solve for all of the transformations jointly.

Unlike the single view point case, when the images are taken from multiple view points one cannot expect all the matches to be nicely aligned. Assigning the same importance to all matches (as was done in [2] for the single view point case) will result in misalignments distributed across the whole panorama. Instead, one would like "important" matches to be well aligned while allowing other matches to have larger errors. This can be achieved by assigning each feature match with a weight indicating its importance. The decision on which features are "important" and the setting of the weights will be described in Section 5.

The objective function of the optimization process is thus a weighted sum of projection errors:
Let u_i^k denote the k'th feature in image i and S_{ij} a similarity transformation between images i and j. Given a feature match $u_i^k \leftrightarrow u_j^l$ the corresponding residual is: $r_{ij}^{kl} = u_i^k - S_{ij}u_j^l$ and the assigned weight is denoted by w_{ij}^{kl}. The error function to be minimized is the sum over all images of the weighted residual errors:

$$e = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} \sum_{k,l \in \mathcal{F}(i,j)} w_{ij}^{kl} f(r_{ij}^{kl}) \qquad (1)$$

where n is the number of images, $\mathcal{N}(i)$ is the set of images with feature matches to image i, $\mathcal{F}(i,j)$ is the set of feature matches between images i and j and $f(x)$ is a robust error function:

$$f(x) = \begin{cases} |x| & \text{if } |x| < x_{max} \\ x_{max} & \text{if } |x| \geq x_{max} \end{cases} \qquad (2)$$

This robust error function is used to minimize the impact of erroneous matches. As suggested in [2] we use $x_{max} = \infty$ during initialization and $x_{max} = 1$ pixel for the final solution. This is a non-linear least squares problem which we solve using the Levenberg-Marquardt algorithm.

Figure 2.a shows an alignment result with equal weights assigned to all feature matches (i.e., $w_{ij}^{kl} = 1$

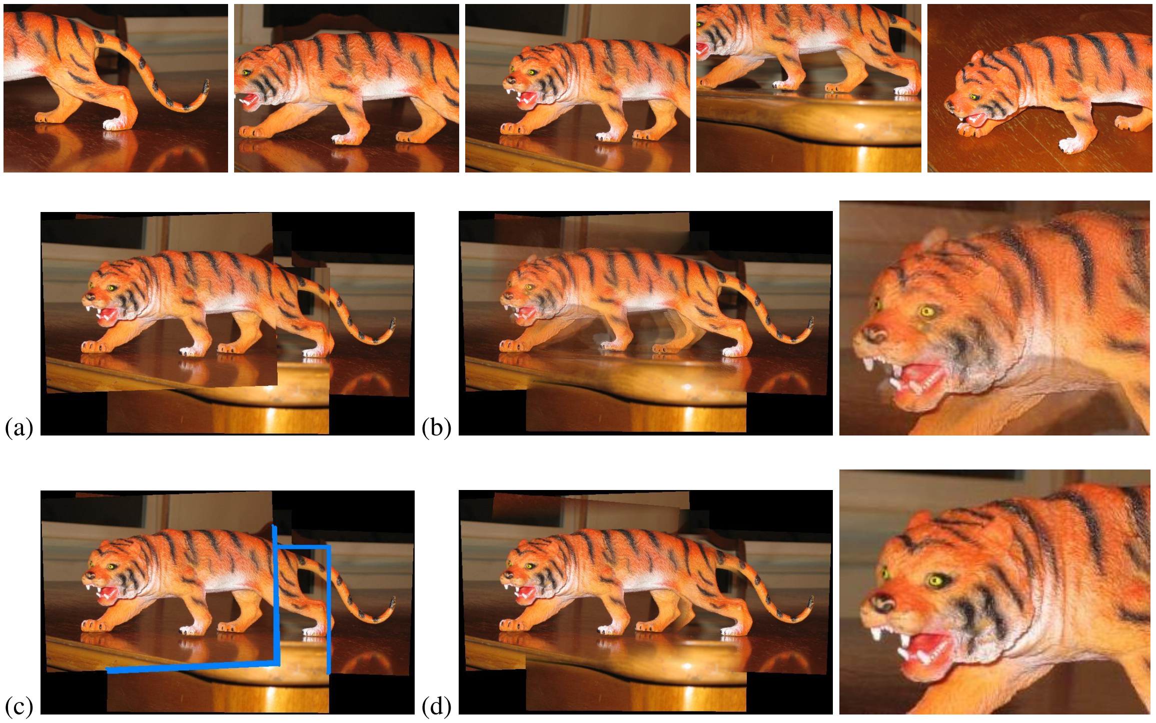


Figure 2: **Panorama construction phases:** (Top row) Input images. (a) Layering the images according to the best order found by minimizing the gradient-based cost. (b) Result of aligning and blending all input images. The tiger's face (enlarged on the right) is blurry (c) Visible image boundaries marked in blue on top of the panorama in (a). (d) Final result after layering and local blending only along visible image boundaries. The tiger's face (enlarged on the right) is now sharp.

$\forall i, j, k, l$). The aligned images were blended using multi-band blending [3]. Parts of the panorama looks sharp while other parts are blurred due to misalignments.

4 Ordering Images

Imperfect alignment will unavoidably result in blurry regions when blending the images. Thus, instead of blending the images we wish to order them into layers such that images placed on top will hide misalignments underneath. This will leave us with visible artifacts only along image boundaries which are not occluded. We will refer to these as "visible image boundaries" (see Figure 2.c). Our goal is to find an order of the images which minimizes appearance inconsistencies across the visible image boundaries.

One can adopt two approaches to order the images:

1. Assign each image to a separate layer and find the best order of layers. This is equivalent to what can be easily done in most image editing softwares, e.g., Photoshop.

2. Select a local order of the images separately in each overlap area. For example one could have image A above B, B above C and C above A in different regions of the composition.

Constructing numerous panoramas manually we found the first option to be sufficient in most cases. We have thus left the second option outside the scope of this paper.

Given an alignment of the images on the panorama plane, finding the best order of images can be formulated as a graph problem. Let $G = (V, E)$ be an undirected graph where each node $v_i \in V$ represents an image and edges connect between images that overlap. A valid order of the images can be represented by an acyclic orientation of the graph edges. The set of all acyclic orientations of the edges of G represents all possible orders of the images. It can be found in overall time $O((n + m)\alpha)$ [1], where n is the number of nodes (images), m is the number of edges and α is the number of acyclic orientations.

We then perform an exhaustive search over all possible orders and select the best one. For each order of the images we compute a cost based on image-to-image consistency measured along visible image boundaries, denoted by $\mathcal{B}$. One can design many such cost functions. We have experimented with three:

1. Sum of gradients across image boundaries: $Cost_{grad} = \sum_{x,y \in \mathcal{B}} P_x^2(x, y) + P_y^2(x, y)$, where P_x and P_y are the horizontal and vertical derivatives of the composition.

2. Sum of color differences between overlapping images: $Cost_{color} = \sum_{x,y \in \mathcal{N}(\mathcal{B})} (I_{top}(x, y) - I_{scnd}(x, y))^2$, where I_{top} and I_{scnd} are the top and second from top images on one side of the visible boundary $\mathcal{B}$ and $\mathcal{N}(\mathcal{B})$ is a region around the boundary.

3. Quality of curve continuation: We first find curves of length ≥ 5 pixels in all images[1] and project them to the panorama plane. We then find the set of curves C which intersect visible image boundaries and are visible in the panorama (i.e., are not occluded by other images). For each such curve $c \in C$ we find the closest curve $\tilde{c}$ on the other side of the boundary. We fit a line to the last 3 pixel-long bits of both curves. Denote by $L(c, \tilde{c})$ the sum of squared distances between the curves and the fitted line. The curve continuation cost is defined as: $Cost_{curve} = \sum_{c \in C} \min(L(c, \tilde{c}), \tilde{L})$, where $\tilde{L}$ is a penalty for curves whose continuation could not be found.

In our experiments we found that in most cases minimizing $Cost_{grad}$ or $Cost_{curve}$ provided comparable results, better than those using $Cost_{color}$. For consistency in the presentation of the paper, all the presented results were obtained by minimizing the gradient-based cost $Cost_{grad}$.

Clearly, for large datasets the number α of possible orders is too large to test all. To overcome this limitation one has to adopt some heuristics. One possibility is trying just a limited number of random orders and keeping the best one. Alternatively, one can start from a small set of random orders and conduct a little search around each one by performing a small number of order flips between images. Another possibility is to compute pair-wise image costs, ignoring the rest of the images and finding the best corresponding order. This can be done by first removing from the original graph G enough edges to destroy all cycles (removed edges are chosen at random out of the edges

[1] We used a software written by the Oxford Visual Geometry Group based on Canny edge detection.

participating in each cycle) and then finding the minimal cost orientation.

Figure 4 shows an example of a composition of 14 images, trying only 100 random orders and selecting the one which provided the minimal cost. The result is imperfect in terms of consistency, yet we find it visually compelling. In all other experiments presented in this paper we limited the number of images per dataset to 7 and applied the exhaustive approach. We believe that efficient search methods do exist, but this aspect is left outside the scope of our current investigation.

5 Iterative Refinement

The approach we adopted layers the images in the panorama so that parts of the images are occluded. This leaves inconsistency artifacts in the panorama only along visible image boundaries. We thus wish for the alignment to be of high quality along those seams while we can afford it to be sloppier in occluded regions. This is achieved by iterative refinement of the alignment and order of images.

Given an initial alignment and order of images we assign weights to feature matches according to their "importance". Matches near visible image boundaries are assigned high weights while occluded matches are given low weights:

$$w_{ij}^{kl} = MAX(\exp^{-MIN(d^2(u_i^k,\mathcal{B}),d^2(u_j^l,\mathcal{B}))/\sigma^2}, \omega) \quad (3)$$

where $d^2(u_i^k, \mathcal{B})$ is the minimum distance between feature u_i^k and the visible image boundaries $\mathcal{B}$. The parameter σ controls the rate of decay of the exponential function and ω defines the minimum weight of a feature. In all our experiments we used $\sigma = 500$ and $\omega = 0.1$.

We obtain a refined alignment by applying the bundle adjustment procedure of Section 3 while incorporating the assigned weights. Given the new alignment the images are ordered again and weights are reassigned according to the result. This process is iterated until convergence. In our experiments we applied 3 iterations. Figures 5 and 7 show the refinement that can be obtained by this iterative process.

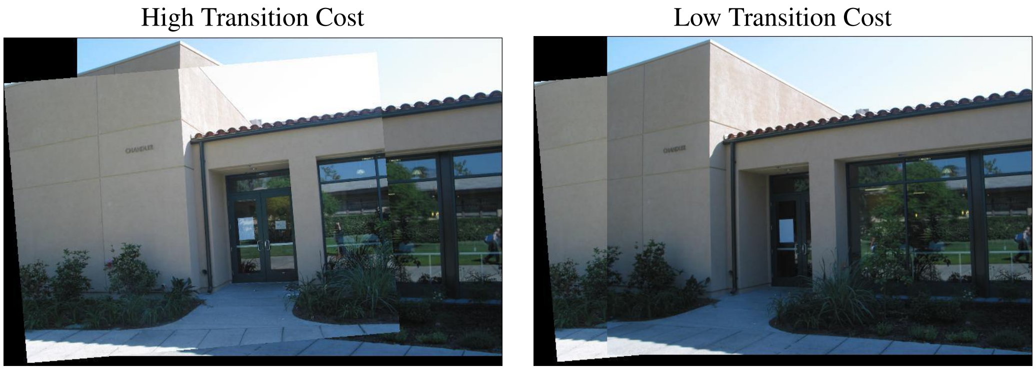
6 Blending

After aligning and layering the images artifacts are left only along visible image boundaries. At this point one can choose between two options, depending on individual taste. The first option is to leave the panorama as is with image boundaries clearly seen, as is commonly done by artists. Alternatively, one can try and remove the visible seams by blending the images. Blending all the images, as is done in the single view case [2], is undesirable since it will make the hidden misalignments appear (see, for example, Figures 2.a,9.a,8.a,8.c). Instead, we apply blending only along visible image boundaries and use only the top and second from top layers. When the alignment quality is high this removes seams while not introducing blurriness, see Figures 2.d,9.b,8.b,8.d. Figure 4 compares the result with and without blending when the alignment is imperfect. We prefer the non-blended result in Figure 4.a, but others may prefer the blended one in Figure 4.b. In our experiments we used the multi-band blending approach suggested in [3].

7 Discussion and Conclusions

In this paper we've shown that stitching images taken from multiple view points is not an impossible task. In many cases nice looking results can be obtained. This was achieved by adding to the traditional geometrical consistency measure a new term which measures consistency in appearance. We modulated the geometrical and appearance costs by what is visible and suggested a search through the space of all feasible and distinct orderings. Finally we have generalized the traditional blending technique to inconsistent picture stacks.

Nevertheless, there are still many open problems. The main difficulty was found to be feature matching. When seen from highly different view points, feature appearance changes significantly and matching of corresponding features becomes more difficult and often fails [9]. This can result in too few matches between overlapping images, or even none at all. Sometimes foreground and background indicate different alignments. This problem may be fixed using stereo and giving priority to the foreground. A possible avenue we plan to explore is allowing some user input to direct and assist the panorama construction in such



High Transition Cost

Low Transition Cost

Figure 3: **Layering Images.** Two different compositions obtained by different ordering of the two images in the picture-set. The composition on the left displays many visible inconsistencies and scored a high transition cost. The composition on the right is smoother and scored a lower cost (see Section 4).

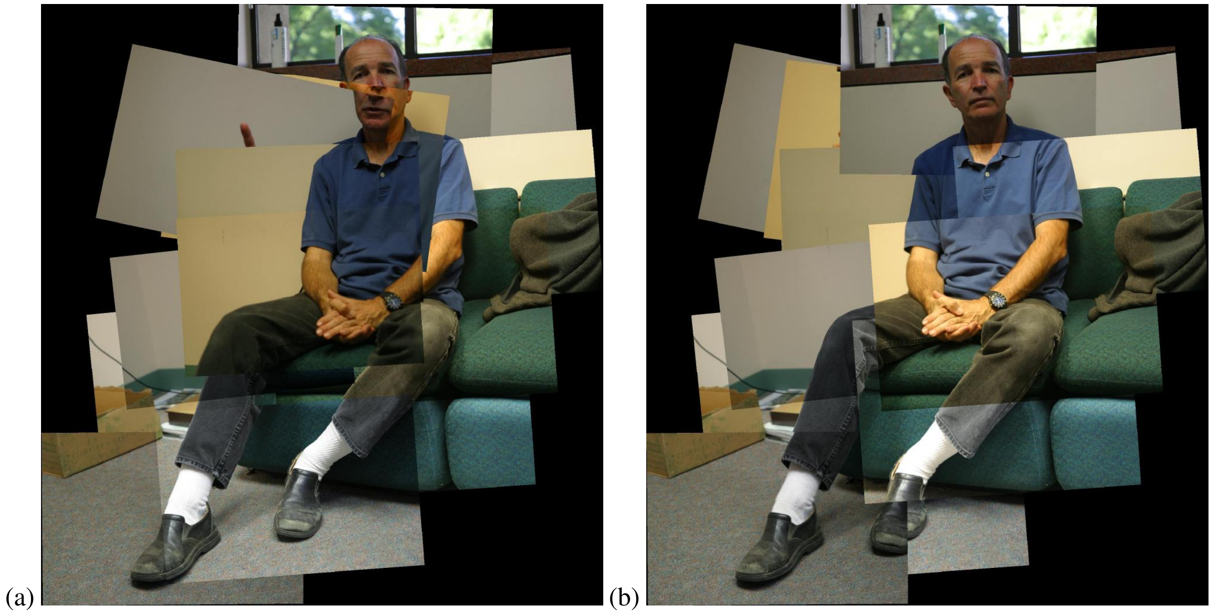


Figure 4: **Large Datasets.** A composition result of 14 pictures showing a man from multiple view points, while moving. Due to the large number of pictures and overlaps between them trying all possible orders will take too long. Instead, we applied the simulated annealing approach of Section 4. (a) Traditional alignment result. (b) Our composition result using iterative refinement of alignment and layering.

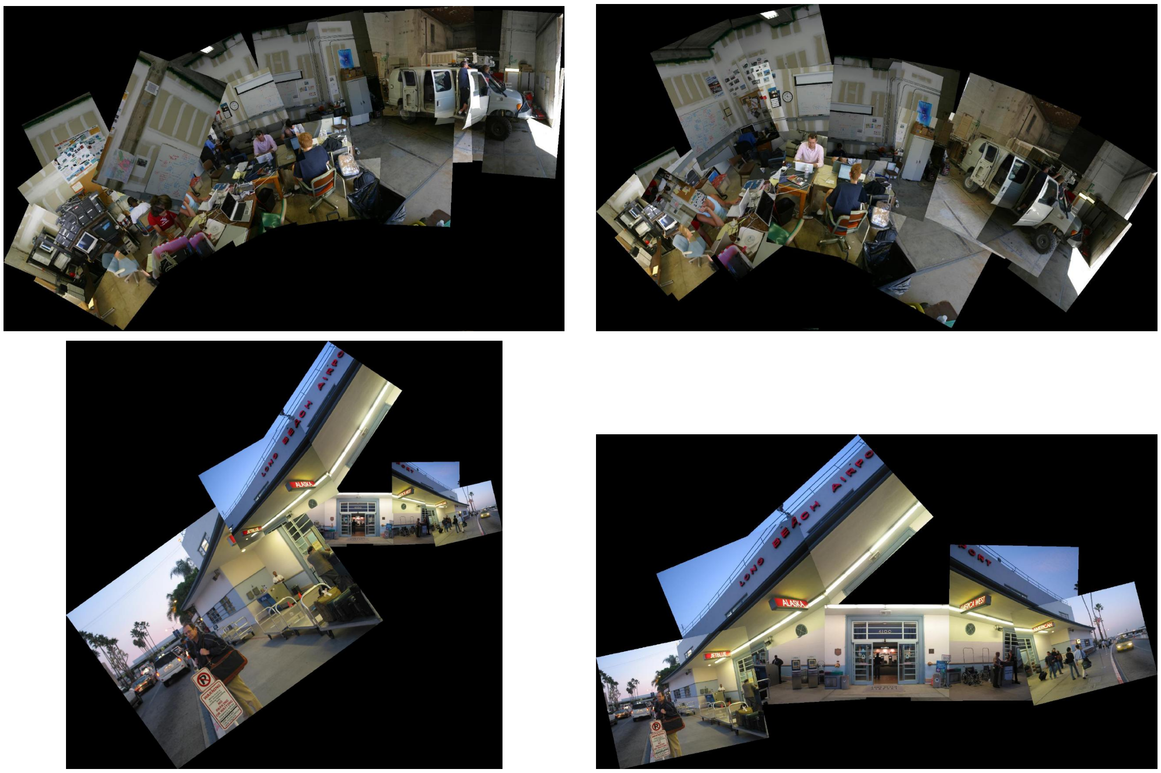


Figure 5: **Iterative refinement.** (Left) Result after a single iteration of aligning and ordering the images. (Right) Final result after iterative refinement of the alignment and order is more visually consistent.

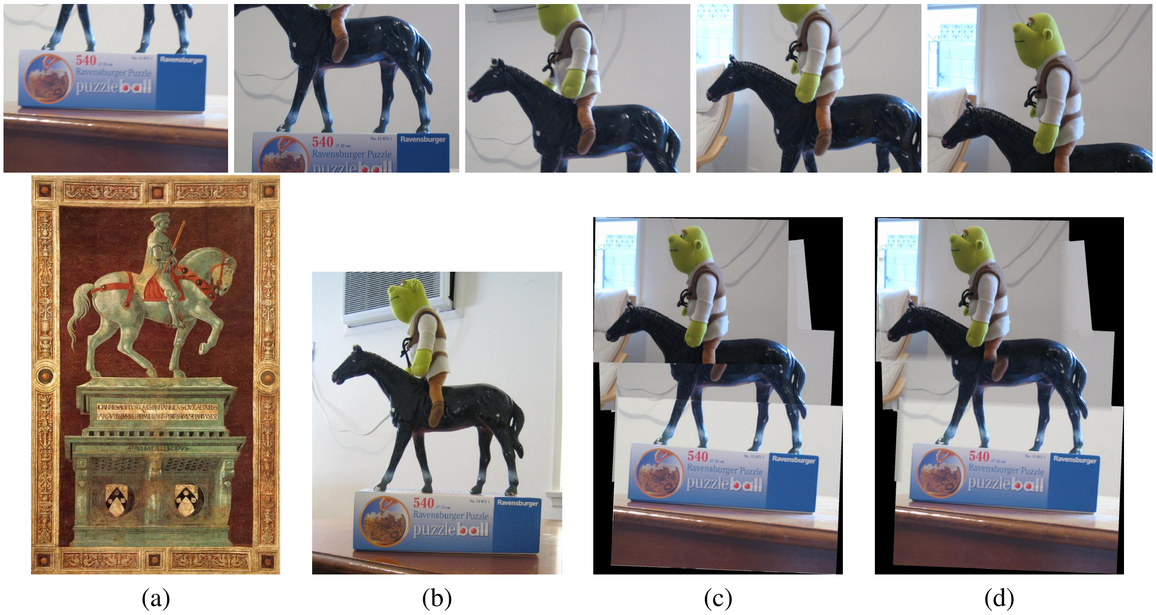


Figure 6: **Incorporating multiple view points:** (Top row) Five pictures of an imitation of the setup in Uccello's fresco (a) taken from different view points. Our "pedestal" was a puzzle box and was thus pictured from a side view to display the text nicely. The horse was pictured both from below, to show its belly, and from above to show the top of its mane. The rider was photographed from a complimenting side view. (a) "Funerary Monument to Sir John Hawkwood" by Paolo Uccello, 1436. Uccello gave the viewer the impression of standing below the pedestal, thus creating a more monumental effect, but at the same time showed the horse and rider from the side providing a better viewpoint of them. (b) A picture of our simulation of Uccello's setup represents what can be seen from a single view point. Viewing the "pedestal" from a side view resulted in viewing the head of the character from below and the nasty grill on the wall behind cannot be avoided. (c,d) Our composition result without and with local blending. As in Uccello's fresco our composition incorporates multiple view points. The "podium" is seen from a direct side view, the horse is seen both from below (showing his belly) and from above (showing the top of his mane) while the rider is again seen from a side view providing a nice portrait of his face.

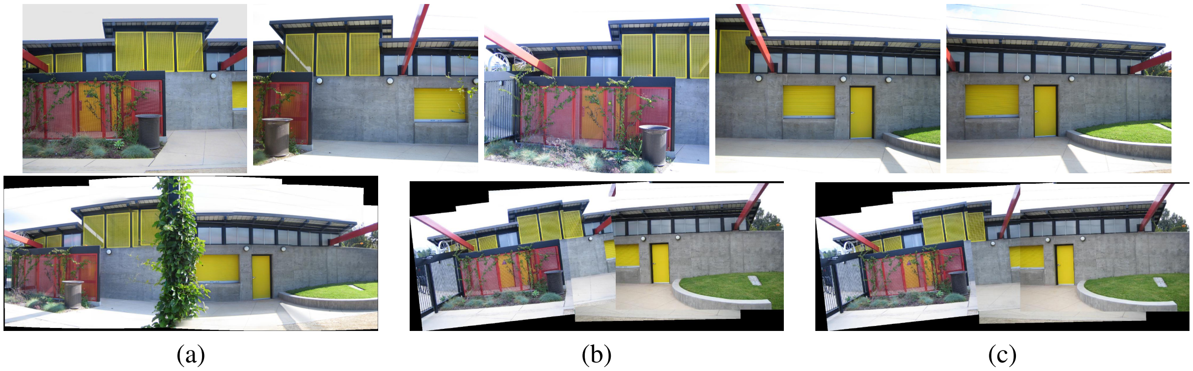


Figure 7: **Avoiding occluding objects.** (Top row) Five pictures of a building used as input for our multi view point algorithm. (a) Single view point panorama of the same building - this used a different set of input images, all taken from a single view point. A pole in front of the building occludes part of it and could not be avoided. (b) Traditioanl composition after a single iteration of aligning and ordering the images in the top row. Extreme discontinuities appear on the yellow window. (c) Our composition results after iterative refinement of the alignment and order is more visually consistent. The occluded pole was avoided by moving the camera and a full panoramic view of the building is obtained.

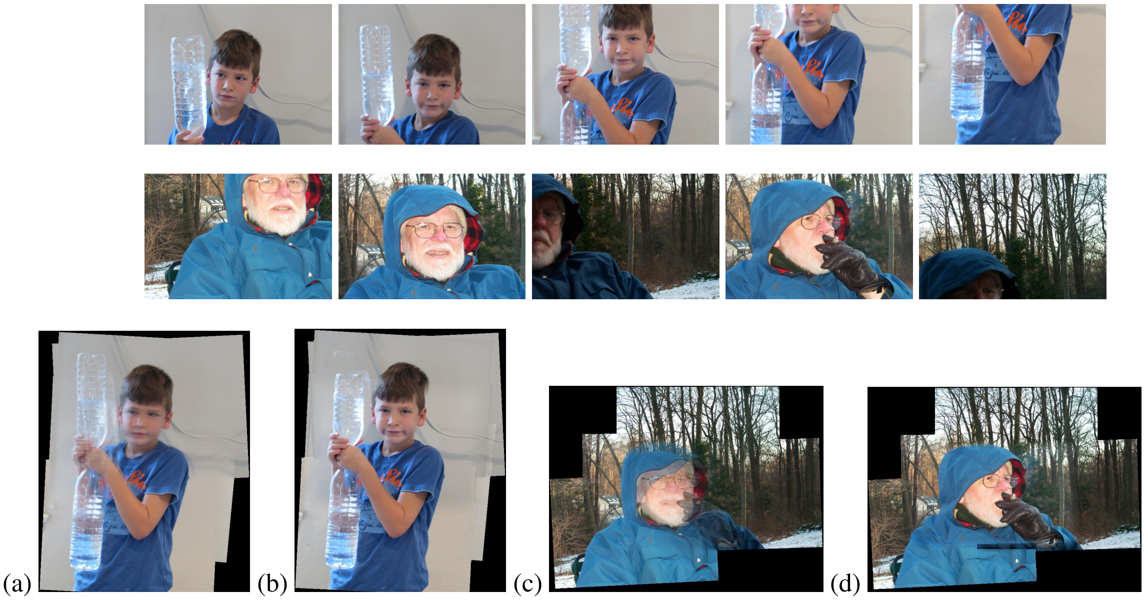


Figure 8: **People.** (Top rows) Two collections of pictures. (a),(c) Results of traditional aligning and blending images of a boy and of a man, respectively. In both cases the people moved their heads, resulting in blurry faces. (b),(d) The corresponding multi viewpoint composition results.

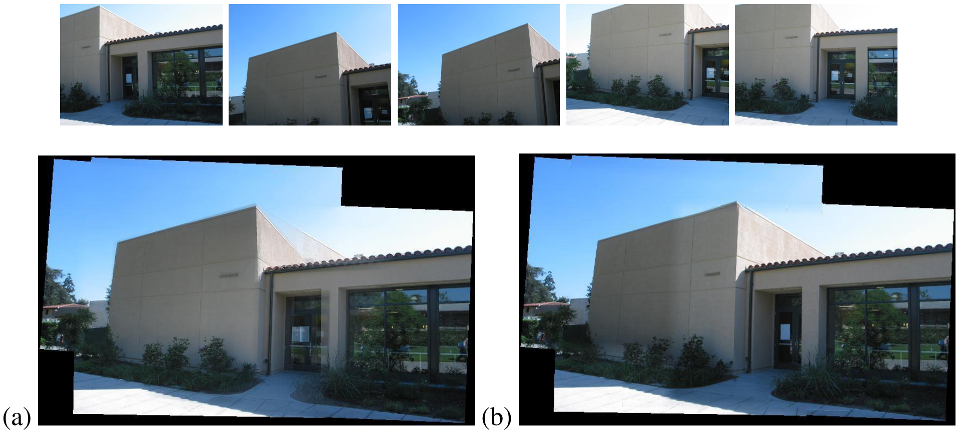


Figure 9: **Local blending:** (Top row) Five pictures of a building taken from different view points. (a) Result of aligning and blending the images. Halos are seen above the building and around the door. (b) Layering the images and applying only local blending produces sharper results.

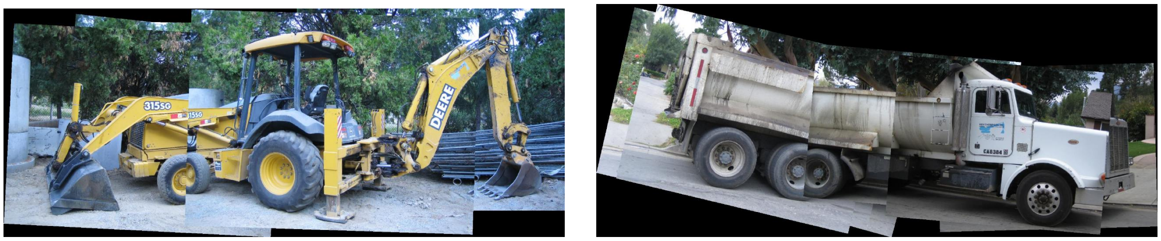


Figure 10: **Compelling Multi View Compositions.**

difficult cases.

8 Acknowledgements

We wish to thank Matthew Brown for providing his Autostitch software and the Oxford Visual Geometry Group for their edge detection code. This research was supported by MURI award number AS3318 and the Center of Neuromorphic Systems Engineering award EEC-9402726.

References

[1] V. C. Barbosa and J. L. Szwarcfiter. Generating all the acyclic orientations of an undirected graph. *Inf. Process. Lett.*, 72(1-2):71–74, 1999. 5

[2] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, pages 1218–1225, Nice, October 2003. 1, 2, 3, 6

[3] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2(4):217–236, 1983. 1, 5, 6

[4] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH*, August 1997. 1

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Proceedings of SIGGRAPH*, August 1996. 2

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1, 3

[7] Y. Li, H. Y. Shum, C. K. Tang, and R. Szeliski. Stereo reconstruction from multiperspective panoramas. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:45–62, January 2004. 2

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 3

[9] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 800–807, 2005. 6

[10] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:279–290, March 2001. 2

[11] P. Rademacher and G. Bishop. Multiple-center-of projection images. In *Proceedings of SIGGRAPH*, 1998. 2

[12] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31(Annual Conference Series):251–258, 1997. 1

[13] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin. Multiperspective panoramas for cel animation. In *Proceedings of SIGGRAPH*, August 1997. 2

[14] L. Zelnik-Manor, G. Peters, and P. Perona. Squaring the circle in panoramas. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 2, pages 1292–1299, 2005. 1

[15] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, June 2003. 2