This paper presents a new combination of techniques to create pleasing and physically consistent image mosaics despite the presence of moving objects in the scene. The technique uses heuristic seam selection in the intensity and gradient-domains to choose which pixels to use from each image and then blends them smoothly to create the final mosaic.

We demonstrate illustrative results obtained by comparing and contrasting our output with that obtained from four representative existing image mosaic systems. One of these is documented in the academic research literature and three are commercially available products. The present algorithm gives comparable or superior results in all examples.

Conventional mosaic synthesis operates by selecting content from one image or another as a function of which side of a dividing line or seam a given pixel lies upon. This paper introduces a new method for selecting the seam by defining a heuristic error metric over pixels with respect to several image properties that define a good mosaic. We assume that the scene is composed of a largely static background with a limited number of moving objects that occupy a minority of the total set of pixels. Ideally, one might wish to explicitly identify and separate these objects and then assure that the image seam is selected to either fully include or exclude an object from the final mosaic, but never select only a fraction of an object for inclusion. Since object-based segmentation is difficult to define unambiguously, we instead seek to avoid it and suggest heuristic functions that strongly penalize seam pixels that are likely to lie in the interior of moving objects. Although user specified compositing hints might simplify the problem, a fully automated solution is much simpler for the user than a solution including manual hints, thus that is the focus of this paper.

A difficulty in creating the seam is that the images being mosaicked may have been taken under different lighting conditions or even with different cameras, creating intensity differences between pixels that contain projections of the same scene points. To cope with this, the images are corrected to have similar exposure parameters using a linear model of bias and gain. Then the seam is calculated based on a combination of differences in the intensity domain and in the gradient domain. If the exposure correction is small, more weight is given to the intensity differences. If the correction is large, the linear approximation is likely to be a
poor approximation and gradient differences are given a higher weight.

Our algorithm performs the mosaicking on successive image pairs in three steps, registration, seam selection and blending.

2. Related work

Image mosaic synthesis can be based on either global image measurements or local features. In either case, these measurements are brought into correspondence with one another in a process called registration. A critical step that distinguishes mosaic synthesis from stereo is the need to merge the images into a single final result. This key process depends on defining a seam along which the merge is defined. The seam can be made nearly invisible by first correcting for exposure differences and then by using a good blending scheme.

Some of the earliest published work on image registration is that of Barnea and Silverman in 1972 [5] and Lucas and Kanade [18].

In the field of exposure correction, the authors of [4] iterate between direct measurement and a linear exposure correction to attempt to register the image. In [26], the image is divided into blocks and each one is individually corrected then smoothed. In [2,15] stitching is done entirely in the gradient-domain to avoid exposure differences. In [7], gain compensation is performed by multiplying the mean value of pixels in each image by a constant to minimize an error function.

After a set images have been registered, a method of blending or selecting pixels in the overlapping regions must be used. The simplest way is to average the pixels, but this leaves visible seams with changes in lighting. Ghosting or parallax can also occur if the camera motion was not purely rotational or if subjects move slightly between each exposure. Some methods designed to improve pixel selection are the median filter [14], center weighting [25], minimum likelihood, optical flow [23] and the over operation [22]. A possible solution to ghosting is to choose a seam in a more intelligent manner and pick pixels from one photo or another based on which side of the seam they fall. One possible seam is a Voronoi diagram [27,21]. This can be combined with blending for a smoother transition [28]. All these methods fail when presented with moving objects that pass through the seam.

The solution to the dynamic object problem is to select a seam based on the contents of the image rather than simply splitting the stitch down the middle [20]. One algorithm for this is to place a seam along the edges of objects in the picture [24]. This fails if a dynamic object is passing over one of the chosen edges at the time the image was taken. A more modern approach is using regions of difference [26]. This technique identifies dynamic objects by checking the source images to see where pixels differ by more than a threshold. Then, to create the mosaic, the object is chosen from one of the images in a pseudo-random way. This algorithm works well, but is easily thwarted by exposure differences. Another technique with excellent results is [3]. This technique requires that the user selects the relative importance of seam and difference and therefore is not suitable for our goal of a fully automated solution.

The closest work to ours in this field is [9]. This paper uses Dijkstra’s algorithm [10] to choose an optimal seam when the source pictures have moving objects. The seam is placed by first creating a difference image of the two registered images, with pixels given higher values if they have larger intensity differences. The seam is then the minimum cost path from one edge of the overlapping area to the other. The formula used by Davis to determine the difference of pixels is also used here. Unfortunately, the algorithm cannot handle illumination changes in the images and therefore is only useful for carefully controlled source images. The reason for this is that the algorithm neglects to compensate for exposure differences and does not use any blending techniques. The work in this paper improves and corrects those omissions and increases the quality of the stitch by considering gradient information. Similar work has been done in [1] and in the field of image quilting [11,16].

3. Algorithm overview

An overview of our algorithm is shown in Fig. 1. First the registration transformation between images (i.e., a homography) is computed, followed by a seam computation and finally a blending of the two images to produce a final result.

A good seam is a seam which divides the final mosaic into regions taken from the source images such that few discontinuities occur along the boundary. An optimal seam falls along the path of lowest intensity in the difference image, producing the minimum discontinuity in the final mosaic. This path will usually avoid objects in motion and regions where lens distortion or parallax effects are noticeable.

An example of input images are shown in Fig. 2.

3.1. Registration

The registration step uses feature-based correspondence to estimate a transformation.

1. Each image has SIFT features identified. These features include a scale, rotation and location on each image, as well as a descriptor.
2. The features are matched by descriptor between images using an approximate nearest neighbor algorithm.
3. The best consistent set of matches under a projective transformation is found using RANSAC [12].
4. The consistent features are used to find the projective transform [13] which will map one image onto the other.
3.2. Optimal seam selection

In the overlapping region, we now decide which image to select each pixel from. A seam is found cutting the overlapping region into two uneven areas, with each region drawing pixels form a different image.

3.2.1. Exposure correction

Truly constant scene illumination is rarely achieved in practice and automatic gain control in the imaging system can lead to apparent changes in lighting. It is therefore necessary to correct for exposure differences. To do this, we use the assumption that the reflectance properties of the scene remain constant in the region of overlap. This allows us to compute the average luminance incident on the photodetector over the overlapping region, and to normalize the input images to keep this constant.

To make the adjustment, a linear approximation is used.

\[
\text{pixel}_1 = \alpha * \text{pixel}_2 + \beta
\]

The gain, \(\alpha\) and bias, \(\beta\), between the images are found using a modified version of RANSAC. The algorithm chooses only the static pixels to calculate exposure correction, as the dynamic regions would skew the information. RANSAC inliers for the exposure correction are shown in Fig. 3.

3.2.2. Intensity

To select the seam, some measure of similarity must be calculated for the regions of the image. This is done on a pixel-by-pixel basis using both intensity differences and gradient differences. The intensity differences \(\delta_i^I\) between the pixels of the two images are calculated, as done by Davis [9] by taking the absolute value of the difference of each pixel divided by their maximum.

\[
\delta_i^I = \frac{\text{abs}(I_1 - I_2)}{\max(I_1, I_2)}
\]

After calculating the difference, a small value is subtracted from every pixel (to a minimum of 0) and then normalized. Doing this removes noise from the pixels and encourages the algorithm to choose a longer path to avoid an object, rather than simply the shortest noisy path.

The intensity differences of the sample images are shown in Fig. 4.

3.2.3. Gradient

If the exposure differences between the two images are significantly non-linear, the exposure correction will not work. In this case, the intensity differences will not be a good measure of similarity and a different metric is needed. To obtain this metric, the images are converted to the gradient-domain. The differences of each pixel \(\delta_i^G\) are found and noise is removed using the same technique as before. The gradient differences of the sample images are shown in Fig. 5.

3.2.4. Synthesis of the techniques

The soft constraints resulting from each of these factors is combined using a weighted sum

\[
\Delta_i = w_1 \delta_i^I + w_2 \delta_i^G
\]

Fig. 2. Example pair of input images.

Fig. 3. Pixels chosen to calculate gain. Non-overlapping pixels are in red, RANSAC outliers are in blue and RANSAC inliers are in green. Dynamic objects are mostly excluded.

Fig. 4. Intensity differences between the two images. Black indicates no difference, white is a large difference.
As is illustrated in Fig. 6, weight selection is an important problem. In this example, a greater weight on the gradient \(w_2\) gives the desired result, whereas in other examples the opposite may be true. The image in Fig. 11, for example, fails to find a correct seam if the value of \(w_2\) is increased (not shown).

The optimal weights for combining these functions are dependent on numerous factors that are difficult to evaluate. However, as the illumination differences between the source images increase, more weight is needed on the gradient image to obtain the desired result. The increased weight is likely due to the linear approximation used in exposure correction, which becomes less valid as the differences become more pronounced. Fig. 10, shows an example of a poor linear approximation. Therefore weights are calculated such that \(w_2\) becomes larger as the gain deviates from unity and the bias deviates from null. These weights will be different for every image pair and are calculated automatically based on the exposure correction.

- \(w_2 = (\text{abs}(\ln(a)) + \text{abs}(b))^2\) is the gradient difference weight. In the presence of a large exposure correction, the linear approximation becomes less accurate and the intensity image is less reliable. Therefore, more weight is given to the gradient image.
- \(w_1 = 1 - w_2\) is the weight of the intensity difference. If the calculated value is negative, a weight of 0 is assigned to \(w_1\).

In order to assure the seam does not cut through an accidental low intensity path, it is prudent to blur the weights of the pixels. An accidental path occurs when the colors in a dynamic object happen to fall in such a way that there exists a path of nearly identical pixels. Usually these paths are surrounded by pixels with high intensity differences and applying a blur will eliminate them. In addition the pixels near the exterior of the image are given higher intensities (inflated) to discourage the optimal seam to trace the edge of the difference image. The amount of inflation can significantly change the location of the seam and it is unclear how to determine the best value.

The difference values for the sample images are shown in Fig. 7.

3.3. Optimal seam selection

As we have noted, a good seam is one that is not visible. This suggests that a seam should either span regions where it will either not introduce a brightness change, or where an induced
brightness change will not be noticeable. Moreover, the seam should not introduce inconsistent scene elements (i.e., by cutting an object in half). These criteria can be largely satisfied by finding the minimum-weight path through the combined difference image. As dynamic objects will have high weighted pixels, inconsistent scene elements will be avoided. Similarly, small brightness changes will be favoured over large ones, making a less noticeable seam.

Finding the optimal seam is simply a matter of calculating the minimum-weight path in the difference image.

\[ \text{Path} = \arg\min_{ij} \sum_{i,j} \Delta_{ij} \]

The path is found using Dijkstra’s algorithm [10], modified for use in Matlab. A matrix (min_cost) keeps track of the weight of the minimum cost path from a starting pixel to every other pixel.

Table 1

<table>
<thead>
<tr>
<th>Program</th>
<th>Automatic stitching</th>
<th>Seam selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoStitch [6]</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Panorama Maker(^a)</td>
<td>Manual image ordering required</td>
<td>Yes</td>
</tr>
<tr>
<td>Panorama Factory(^b)</td>
<td>Seven manual point correspondences required</td>
<td>No</td>
</tr>
<tr>
<td>Panavue(^c)</td>
<td>Single manual point correspondence required</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\(^a\) http://www.arcsoft.com/products/panoramamaker/.
\(^b\) http://www.panoramafactory.com/.
\(^c\) http://www.panavue.com/.

Fig. 9. Stitched photos. Note that (b), (d) and (e) use optimal seam selection to obtain similar results, whereas (a) and (c) do not, resulting in a translucent girl.
At each iteration, every pixel is tested against each of its neighbors to see if its neighbor's cost plus its own weight produces a smaller value than its current cost. If so, the value of that pixel is updated. This continues until no updates are made. Only the pixels that recently changed and their neighbors are checked in each loop.

Fig. 10. Stitched pair of photos of a difficult indoor scene. One is taken with a flash and the other without. Panorama Factory was unable to match the photos together. AutoStitch has trouble blending the girl in the two photos. Panorama Maker shows a clear seam due to the difference in lighting and cuts through a section of the girl's head. Panavue did not properly transform the photos and the result is poor. The present implementation stitches the images together nicely.

Fig. 11. Stitched pair of photos of a difficult lab scene. AutoStitch was unable to match the photos together. The other applications have a poor choice of seam. Only the present implementation handles the moving person properly.
Starting at the potential end point with the lowest cost, the path is then retraced back to a starting point using the difference matrix and the weights matrix.

The optimal seam for the input images is shown in Fig. 8.

3.4. Blending

Seam selection determines the borderline between images to be merged, but the pixels on either side of the seam will usually still exhibit some degree of inconsistency. To minimize this saliency, we can post-process the local image content as the images are blended. The blending is done using a multi-resolution spline [8]. The multi-resolution technique was chosen as it preserves high frequency edges, which avoids ghosting and complements the optimal seam selection well. The spline is outlined below.

1. Create a mask image using the optimal seam. All pixels on one side of the seam are set to white and on the other side to black.
2. Run bandpass filters on both of the source images and the mask image to obtain a fourier decomposition.
3. Blend each band of image A with the corresponding band of image B using the corresponding mask for that frequency band. Lower frequencies have grayscale pixels in the mask and will have a weighted average based on the level of the gray. This allows the low frequency components to blend more smoothly than the high frequency components.
4. Finally, sum the images in each frequency band to create the final blended image.

The number of frequency bands to use is chosen based on the size of the overlapping region. As each additional band doubles
the size of the blended region, the number of bands is chosen to maximize the amount of the overlapping region used in blending, as more bands will create a smoother blend.

Blending the images in this manner allows low component information to be dispersed and blended over a large radius, making a smooth seam even in the presence of exposure differences. This technique complements the optimal seam selection as the seam finds areas which have nearly identical high frequency elements, but potentially varying low frequency elements.

The result of the process leading to a blended mosaic of the two sample images can be seen in Fig. 9e.

3.5. Adding more images

Our approach computes a mosaic from an ensemble of images by performing successive pair-wise combinations and gradually building up a composite image. The images are first selected by the pair that has the best match based on the ratio of RANSAC inliers to outliers found in the feature matching step. The third image selected to blend onto the mosaic is the one having the best inlier to outlier ratio to either of the two images already in the mosaic. Additional images are stitched to the mosaic using this selection criteria. A third image stitched to the previous result can be seen in Fig. 13.

Fig. 13. Three photos stitched together. The photos have all been cropped.
4. Results

Our algorithm was compared with four other available panoramic applications listed in Table 1. The results can be seen in the following figures. These examples were chosen to illustrate some of the real-world situations for which our algorithm was designed.

- **Fig. 9:** This example is a simple case. The two source images from Section 3 were provided as input. There is substantial overlap between the source images and dynamic elements take up only about 20% of the region of overlap. All three algorithms which use optimal seam selection perform well.

- **Fig. 10:** This example illustrates exposure differences. The motion in the scene also projects to a larger portion of the overlap region between the images. One of the input images was acquired using a flash and the other was taken only with natural illumination. This is a much more difficult problem and only the present method produces an inconspicuous seam. To obtain a satisfactory result, exposure correction as well as optimal seam selection must be used. Not all seam selection methods will work.

- **Fig. 11:** This example illustrates a scene with a large dynamic object. Panorama Maker, Panorama Factory and Panavue all make a noticeable cut through the subject’s body. The present application chooses the pose from the left image rather than the right and as a consequence, makes a seamless stitch.

- **Fig. 12:** This example shows how the algorithm handles noticeable parallax. Most of the programs were unable to even register the images. The present algorithm finds a nice seam to blend the images together. This seam was very dependent on how the edges of the difference image were inflated. The reasons for this are unknown.

- **Fig. 13:** This example shows stitching with more than two images and to show a scenario where an optimal seam does not exist. In this image there is a car which is partially in the frame, making it impossible to create a ‘perfect’ stitch as there is no seam which either includes or excludes the car.

To run the algorithm on the images shown in Fig. 2 took 56 s. The images were each 300 × 400 pixels. Of this time, 16 s was spent finding the optimal seam. The tests were performed on an Intel Pentium 4 CPU @ 3.20 GHz.

5. Discussion

A technique has been presented to automatically stitch and blend together images with dynamic elements. It partially solves the problems of moving objects, small parallax, illumination changes and lens distortion. The algorithm was compared with pre-existing solutions from both the commercial field and the research literature, and found to produce comparable or superior results to all of them.

The algorithm struggles with objects for which the entire object can neither be included or excluded. One method of dealing with this would be to attempt to identify when the optimal seam cuts through an object. If so, a more severe cross-fade blend could be used to create a smoother transition.

The algorithm handles small parallax artifacts fairly easily, and sometimes works well with large parallax. Although to perfectly handle this problem full 3D recovery and registration is required, it may be possible to improve upon the algorithm’s handling of the problem. One problem which seemed to occur was that in the presence of parallax-induced inconsistencies, where there is no seam with perfect overlap in the image, the algorithm often chooses an optimal seam along the border of one of the images. This was discouraged by introducing an ad hoc penalty term along the border pixels. A more principled approach to this problem would be attractive.

With suitable optimization, the mosaic synthesis process can be accomplished very rapidly. For example, some techniques can approach real-time speeds [7], albeit without operations such as the optimal seam selection we employ. While our work does not currently address practical deployment issues, several interesting optimizations may be possible.

The choice of the formula to weight the gradient and intensity difference images was motivated based on empirical observations, but more experimentation is needed to determine which factors should contribute and how.

References


artifacts in image mosaics, in: Proceedings of the IEEE Conference on 

Multiperspective panoramas for cell animation, in: Proceedings of the 24th 
Annual Conference on Computer Graphics and Interactive Techniques, 1997, 
pp. 243–250.

[28] Y. Xiong, K. Turkowski, Registration, calibration and blending in creating high 
quality panoramas, in: Fourth IEEE Workshop on Applications of Computer 