Mosaicking Images with High Motion Parallax with Application to Video Compression

E. Khameneh  
Sharif University of Technology  
khameneh@ce.sharif.edu

S. Kasaei  
Sharif University of Technology  
skasaei@sharif.edu

Abstract: Image mosaicking has been a focus of attention of many researchers in recent years. Mosaicking methods which exist today are merely unable to construct mosaics from images taken with large motion parallax. The idea is resembles some kind of layered mosaicking. The proposed method uniformly distributes the parallax mismatch between consecutive frames in the whole mosaic. In this paper, we have shown the results of the developed algorithm on two consecutive frames, as well as the result obtained on a set of frames. Although the resulted mosaic gets a bit blurry by using a set of frames with high motion parallax, it can be completely useable for video compression where the mosaic is used to compute the residuals. The proposed algorithm is much faster than the existing methods which tend to have computation time of several seconds to several minutes. Besides, a multi-resolution version of the algorithm is introduced will lessens the computation time considerably. The obtained results show the efficiency of the proposed algorithm.

1 Introduction

Image mosaicking is about stitching small images together to produce a larger mosaic image. Methods of image mosaicking vary widely depending on the application it is involved in. The applications include video surveillance, remotely sensed and medical image analysis, virtual reality, video compression, etc.

The main work done in this field is using parametric 2-D transformations such as affine, bilinear, and projective transformation in registering the corresponding images together [21,22]. These methods can only be used in the following two cases:

1. An arbitrary 3-D scene viewed by a camera at a fixed location free to rotate about its center of projection.
2. A planar or very distant scene viewed by a camera free to move arbitrarily.

When dealing with applications such as mosaicking video frames, it becomes necessary to consider motion parallax. The camera translational motion when capturing images from a scene with varying depths results in parallax between consecutive frames [1, 4, 5, 6, 12]. Although other methods such as multi-perspective mosaicking exist [8], layering the images is the only feasible method for mosaicking images with high motion parallax; where each layer represents a different depth in the scene. This type of representation is sometimes called 2.5D scene representation. No linear transfer function can register the two images with motion parallax completely well. Obtaining the non-linear transfer function requires extracting depth information for each pixel [9]. This method has been used in [8], which leads to very high computation time. It is impossible to calculate the depth map if the displacement between the two images is large. Furthermore, the gaps or holes in the resulted mosaic are also unavoidable [4]. These gaps can be closed by using the closing morphological operator which leads to additional computation time. It is worth mentioning that none of these methods are able to cope with large motion parallax. This fact is observed from the sample images used for making mosaic images.
In [1], a third image is used to extract the depth information. These methods are too complicated to be used in video compression. This paper has aimed at a more efficient method for mosaicking images with motion parallax, which can be used in video compression. The rest of this paper is organized as follows. Section 2 gives the reader a background of why the method used has acceptable results. Section 3 delves into the implementation details of our work. In Section 4 we have illustrated some experimental results, and Section 5 concludes the paper.

## 2 Background

As it has been discussed earlier, layered image mosaicking is the only feasible method for dealing with large motion parallax. Therefore, we have proposed a new type of layered mosaicking in this paper. The difference of our method from the existing methods is our layering scheme and the method of combining the layers to form a complete mosaic image. Two images with motion parallax are illustrated in Figure 1. As can be seen in this figure, by a certain amount of displacement of the camera, the objects indicate different displacements, proportional to depth of them. This phenomenon is observed by looking at the door handle which is far from the camera and the nearer objects such as the calculator. Feature detection and tracking is a widely known method for registering images [23]. It offers more computation speed and accuracy [16]. The feature detector and tracker used here is based on the method developed by Lucas, Kanade, and Tomasi [9, 10]. This feature tracker was mainly developed for finding depth information in stereo images. A feature is detected in parts of the image which constitute more information rather than other parts. These features are usually the parts of images with high spatial frequency contents (such as corners, which have both horizontal and vertical edges.)

Here, we have offered a method which distributes all of the extra information of motion parallax uniformly throughout the whole constructed mosaic image. It will be shown that this method works best with small movements of the camera with large motion parallax or small motion parallax with larger movements of the camera, making it more ideal for video compression applications. Note that there is a trade off between the amount of parallax in the images and the amount of camera displacement for achieving a high quality mosaic.

![Figure 1: Two images with parallax (compare the door handle and the calculator displacements).](image1)

The main idea behind our method is to estimate the layers of the scene being observed and somehow combine the layers into a single mosaic in a best effort manner. Therefore, we will not eliminate any information residing in the images, but we fuse them together making a mosaic incorporating the most information possible from the input video frames. This may result in ghosting effects and doubles as indicated by most of the authors, but here we intend to minimize those ghosting and doubling effects, and not to eliminate them. Eliminating ghosts and doubles can be described as eliminating some of the useful information residing in the images. Therefore, we have made those ghosts to spread out in the whole mosaic image. As we will see, with large motion parallax this may lead to a mosaic image not very suitable for visualization but very suitable to video compression.

In order to achieve this goal, we have adopted a certain layering method based on feature detection and tracking. To layer the scene, we have grouped the tracked features into different layers, by simply calculating the relative displacement of the features. Therefore, features with similar displacements are grouped and classified into a single layer. Some threshold or a fuzzy classifier may be used to obtain better clusters belonging to the same layer. Afterwards, we construct the partial mosaics based on different sets of obtained features. These partial mosaics incorporate doubles and ghosts in areas of the image not belonging to the corresponding layer represented by the partial mosaic. Each partial mosaic represents the mosaic obtained by the whole images in one of the layers. The next step in creating an overall mosaic image is combining the partial mosaics together. Many combining methods may be used to achieve a final mosaic. We have used a simple and yet very effective method of combining these partial mosaics. We have performed a weighted average between all of the partial mosaics to form our final mosaic. The weights are based on the number of features residing in the corresponding layer. The more the features in a layer, the more it is
considered in the final mosaic. Features are parts of the image with higher intensity changes. Therefore the layers with more features are taken into account more than those with lesser features. This will make the final mosaic visually acceptable. Averaging has an affect in reducing the ghosts and doubles. Ghosts and doubles are parts of the partial mosaics normally differing from each other. Therefore, averaging makes those ghosts fade out gradually, so much like how noise fades out when averaging noisy images.

The result is a mosaic image which has eliminated the ghosts and doubles, and somehow distributed the parallax error in the entire image, making it less observable. No gaps or wholes are produced in this method, so no closing operation is needed. The final mosaic has very acceptable quality for sequences with small displacements, or small parallax.

3 The Algorithm

As stated earlier, our algorithm uses the Kanade-Lucas-Tomasi (KLT) feature tracker. The steps involved in constructing the final mosaic are as follows:

Feature detection and tracking using KLT: The KLT feature tracker is very robust when dealing with stereoscopic images, where large amount of parallax is involved. This feature tracker selects \( N \) features and tracks them in the next frame. Figure 2 illustrates the features detected and tracked by KLT.

Features classification: The tracked features are classified into different layers according to the amount of displacement of each feature.

Building partial mosaics: The partial mosaics are built by using the displacement information for each set of features corresponding to the same layer. We have combined the two images together by using the following equation:

\[
m_l(x, y) = (1 - w)v_1 + wv_2
\]

where \( m_l(x, y) \) is the partial mosaic image for layer \( l \), \( w \) are the varying weight, \( v_1 \) and \( v_2 \) represent the values of intensity in the two images which correspond to each other. It makes the transition of the two images together as smooth as possible. Wavelets can also be used to combine these images which have been shown to have more natural results [15]. Two examples of these partial mosaics are illustrated in Figure 3. The partial mosaic on the left corresponds to the nearest layer (the calculator), and the partial mosaic on the right corresponds the farthest layer (the wall). It is observed that ghosting and doubling occurs at other layers.

Combining partial mosaics: Finally, the partial mosaics are combined by using the following equation:

\[
m(x, y) = \frac{\sum_{l=0}^{L-1} w_lm_l}{\sum_{l=0}^{L-1} w_l}
\]

where \( m(x, y) \) is the final mosaic and \( w_l \) is the weight assigned to layer \( l \), which is proportional to the number of features residing on that layer.

The following method is used for mosaicking a series of video frames for achieving better results. The video frames are combined two by two. This method works on \( 2^n \) frames. Suppose we want to combine 8 frames together:

• Combine frames 1 and 2, 3 and 4, 5 and 6, and 7 and 8.
• Then, combine the mosaics 12 and 34, and 56 and 78.
• Finally, combine mosaics 1234 and 5678 to form the final mosaic.

The next section will demonstrate the results obtained from applying these algorithms on some typical input images.
To decrease the computation time, we have offered a method to perform the same operation in a lower resolution:

1. Take the image to a lower resolution.
2. Detect and track the features in that resolution.
3. Take the tracked features back to the original resolution.
4. Perform the mosaicking process.

We will see the performance improvement of this kind of mosaicking in the following section.

### 4 Experimental Results

It seems that the only algorithm which aims at high motion parallax and is used for visualization is the algorithm described in [8]. This algorithm constitutes the following steps:

1. Pre-processing the frames.
2. Calculating optical flow, by using the Lucas-Kanade algorithm, which only works up to 5 pixels per frame.
3. Mosaicking the layers independently.
4. Combining the layer mosaics.
5. Performing the closing morphological operator to close the gaps resulted.

According to the outlined steps, specially the second and fifth step, it has a high running order depending on the number of pixels residing in the frames and the size of mask used for each pixel. The performance of this algorithm was not discussed. Other papers such as [1, 4, 5, 12] do not offer any performance evaluation on images with high motion parallax. Therefore we will only offer performance results for the proposed algorithm.

We tested it on a sequence of 8 images shown in Figure 4. This image sequence has very large motion parallax.

The whole mosaic consisting of these frames is calculated by the method outlined in the previous section. The steps involved in this process are illustrated in Figure 5. The more mosaics are combined, the more motion-like blur is resulted from the parallax. The proposed algorithm spreads the extra information resulted from motion parallax throughout the final mosaic, therefore eliminating any ghosts or doubles. It is observed that the mosaics created from 2 and 4 frames meet the quality needed for visualization purposes. The proposed algorithm produces acceptable quality images for 2, 3, and 4 frames. The quality for visualization degrades as the number of frames increases. This is due to the fact that the extra information added by parallax gets increased by increasing the number of consecutive frames. But, this is of no concern to the application of mosaicking for video compression. In video compression, the mosaic image is considered as a reference image, which is used to calculate the residuals of each frame for coding purposes. It may be also more suitable for these kinds of applications, as it includes information from all of the frames in the final mosaic.

### Table 1: Running time of the proposed algorithm. (in milliseconds)

<table>
<thead>
<tr>
<th>No. of Features</th>
<th>30</th>
<th>50</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLT Process</td>
<td>426</td>
<td>464</td>
<td>478</td>
</tr>
<tr>
<td>Stitching Process</td>
<td>34</td>
<td>53</td>
<td>65</td>
</tr>
<tr>
<td>Total Running Time</td>
<td>450</td>
<td>517</td>
<td>543</td>
</tr>
</tbody>
</table>

![Figure 4: Input image sequence with very large motion parallax.](image)

![Figure 5: The steps involved in the process.](image)

![Figure 6: The result obtained from two frames with smaller motion parallax.](image)
The running time of this algorithm is summarized in Table 1 for different number of features. These results were obtained on an AMD Turion 64 ML-30(1.6GHz), 1GB RAM. Each frame is a 320×240 image. The major time of the process is taken up by the feature detector. This means that different feature detectors will result in different running times. Thus, by using a much more efficient feature detector, the running time can be reduced significantly. The number of features has little effect on the running time. Only 30 features have been used to obtain all of the results given in the present section. The quality of the mosaic has little dependency to the number of features used. The result of the algorithm after applying it to frames with smaller motion parallax was very encouraging. Figure 6 shows two frames with smaller motion parallax which have been used to obtain a mosaic image. It shows the result of the algorithm on an office like scene. The occlusions due to parallax are observable near the front legs of the table and the front left edge of the wall. Note that the resulted mosaic image lacks any inconsistency and contains no doubles or ghosting in it.

We also tested the running time of multi-resolution algorithm, which was considerably faster than the normal algorithm. The results are summarized in Table 2.

**Table 2: Comparison of the running time between the two algorithms (in milliseconds).**

<table>
<thead>
<tr>
<th>Calculation Type</th>
<th>Simple</th>
<th>Multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLT Process</td>
<td>416</td>
<td>109</td>
</tr>
<tr>
<td>Stitching Process</td>
<td>34</td>
<td>33</td>
</tr>
<tr>
<td>Total Running Time</td>
<td>450</td>
<td>142</td>
</tr>
</tbody>
</table>

3 Conclusion

In this paper, we presented a very robust method for mosaicking images with high motion parallax. The proposed method uniformly distributes the parallax mismatch between consecutive frames in the whole mosaic. Although the resulted mosaic gets a bit blurry by using a set of frames with high motion parallax, it can be completely suitable for video compression where the mosaic is used to compute the residuals. The proposed algorithm is much faster than the existing methods which tend to have computation time of several seconds to several minutes depending on the processor and image size used. By choosing a better layer classification technique, better results may be obtained.

Acknowledgement

This work was in part supported by a grant from ITRC.

References


