1. Introduction

Image stitching is commonly used in many different applications. One such application could be creating panoramas or high resolution images. To create seamless images is often very computationally demanding. On the other hand, the technological evolution of the latest years has created a customer demand for these algorithms to be used in handheld devices, such as cameras and cell phones. The methods need therefore to be faster in order to work in near real-time on mobile devices. Using one of the simpler algorithms often results in degrading artifacts, such as artificial seams or blurred regions. To favor seamless blending and clarity in the blending areas, while keeping a low computational cost, is the main motivation for this report. In this report the implementation of a fast image blending algorithm, based on the watershed algorithm and graph cuts, is presented [5]. The images are first aligned using SURF-matches.

2. Background

The methods available for image blending can be divided into two categories, the first being transition smoothing and the second being optimal seam finding [1].

Transition smoothing methods try to minimize the seam between the images by smoothing the edges of the images, commonly referred to as alpha blending or feathering. The largest disadvantage of transition smoothing is that the method in it most naïve form creates blurry areas. Recently, several methods using multi-resolution blending, such as using wavelets and gradient domain blending, have been presented [5]. However, generally the methods require finding a least-square solution of a Poisson equation which is very computationally expensive [6].

Optimal seam finding methods try instead to place the seam where the intensity difference between the two images is as low as possible. The method implemented in the report belongs to this group. There are methods trying to match the intensity on pixel level, while other methods instead divide the images into segments and match the segments between the images. Pixel level matching methods often give better results than region matching methods, but are more computationally demanding [5]. One of the largest challenges using region matching is to find good regions.

2.1. Watershed segmentation

In watershed segmentation one regards the image as a topological map where darker areas correspond to valleys and holes and brighter areas correspond to ridges and tops. A water source is placed in all regional minima. The water level rises and starts to flood the landscape. When water from different sources meet a barrier is built. The resulting barriers will then define the different segments [8]. One example of a watershed segmentation is shown in Figure 1.

![Watershed regions](image)

Figure 1. Watershed regions

2.2. Graph cuts

In this report we try to find the minimum cut in a weighted s-t graph. A graph is a mathematical representation of objects (vertices or nodes) connected by links (edges or lines). Each link in a weighted graph is associated with a weight. In an s-t graph there are a source (s) and a sink (t). When searching the minimum cut one
tries to find the cut that separates the source from the sink to the lowest cost, i.e. the lowest total sum of the weights cut [3]. An example of a graph and its minimum cut is shown in Figure 2.

![Figure 2. Examples of a graph its minimum graph cut. The total cost is 8 (2+4+1+1).](image)

3. Method

3.1. Image alignment

To align the two images, SURF descriptors [2] were calculated in the images, shown in Figure 3. SURF was chosen since it is faster than, for example, SIFT and often also more robust. The descriptors were matched by considering the Euclidean distance, i.e. a descriptor in the first image was assigned to the descriptor in the second image being the closest with respect to Euclidean distance. Outliers were then removed by running a RANSAC algorithm [4]. Iteratively, four matches were randomly chosen and used to estimate a projective transformation between the images. This was repeated 500 times. The best transformation, i.e. the transformation that could transform the largest number of points correctly, was used to transform the second image into the first one, see Figure 4.

![Figure 3. Corresponding SURF descriptors in the first and second image.](image)

![Figure 4. The second image transformed.](image)

3.2. Segmentation of the intersection

After alignment of the images the inverted intensity difference (Figure 5) between the images was calculated, according to the following formula:

\[
D(x, y) = 255 - \max(I_1(x, y) - I_2(x, y))
\]

where \(I_1\) is the first image and \(I_2\) the second image. The watershed segmentation was then performed on the inverted difference image, see Figure 6. By segmenting the difference image, regions which are the least similar are aggregated. When trying to find the best seam those areas should be avoided.

3.3. Graph cut labeling

The source and sink of the problem were defined as the non-overlapping regions from the first and second image, respectively. The segments from the watershed algorithm were set as the nodes in the graph. The total sum in the difference image of the boundary pixels between two segments was set as the weight between the segments. Using a max-flow algorithm the minimum cut was found. The minimum cut was then the opti-
4. Results

Pictures from two different settings have been used to evaluate the method presented. The method has been compared to a simple feathering method, where a vertical seam is used and the border areas has been blurred in order to decrease the visibility of the seam. The first set of images was taken using a cell phone camera out through the author’s office window. Note that there is a difference in exposure between the two images. The second set of images is two frames from a film clip found on YouTube [7]. In the first comparison (Figure 9, 10, 11) one can clearly see the seam for the simple method, while it is much harder to spot the seam for the method presented herein. In the second comparison (Figure 12, 13, 14) one can clearly see that the surfer has been cut in half using the simple feathering method. Even though there are differences in the water texture between the images the method presented in this paper found a seam that is not visible. Overall, one can conclude that the method presented is successful in creating seamless image blendings.

References

**Figure 9.** Images taken with a cell phone.

**Figure 10.** Result from using feathering.

**Figure 11.** Result using graph cut stitching.
Figure 12. Images downloaded from YouTube.

Figure 13. Result from using feathering.

Figure 14. Result using graph cut stitching.