Copy Move Detection

Anthony Sutardja, Kevin Tee

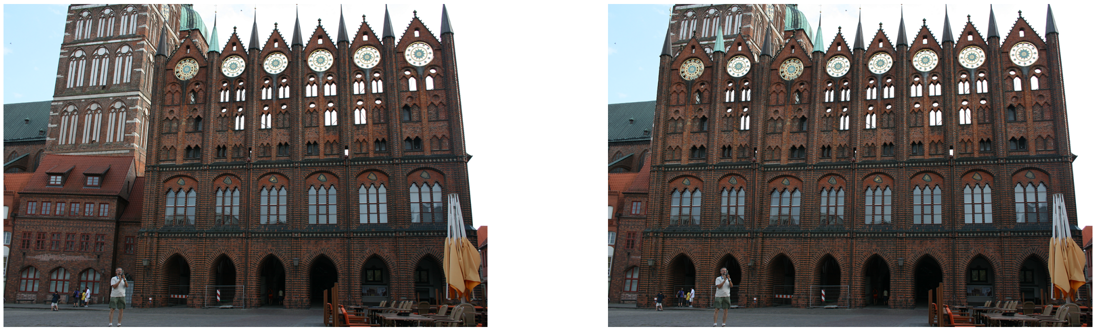
Department of Computer Science
University of California, Berkeley
CS294-26 Final Project

{anthonysutardja,kevintee}@berkeley.edu

Abstract: "Copy-move" forgery, in which a portion of the image is resampled to another region, is a common photo manipulation technique used to alter picture content. Previous work in detecting this type of forgery has shown inaccurate results and inefficient processing time. We propose a combination of methods that results in fast and more accurate copy-move detection performance. Our method finds invariant features within the image and looks for matches using a modified version of nearest neighbors on oriented patches of the images. Potential corresponding matches are filtered by whether they fit into a set of transformations given by RANSAC.

Introduction

Photo manipulation tools have been becoming more powerful and accessible to use than ever before. Almost anyone can open up their favorite photo manipulation tool, like Adobe PhotoShop, and change the image to enhance and alter it. As these techniques get more and more



(a) Original image (b) Copy-moved image

Figure 1: An example comparison of a copy move forgery where a portion of the image has been replaced with buildings sampled from the same image.

advanced, the distinction between what is real and what is been crafted is becoming harder to distinguish, which means that the human eye has difficulty interpreting which images are authentic.

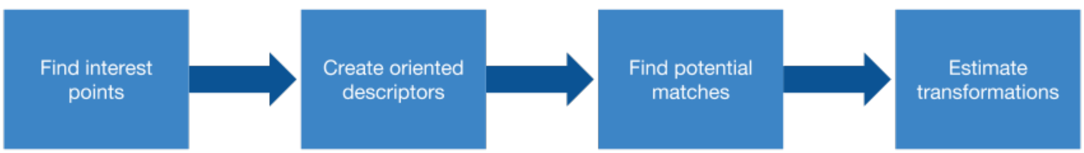
In this paper, we examine the detection of a particular type of image forgery known as copy-move forgery. Copy-move forgery is when portions of the image are resampled to another part of the image with the intent to change the photo's meaning and context. An example of such an attack is replacing a portion of the image with more buildings as seen in Figure 1 [3].

There are many existing methods that attempt to detect copy-move forgery. However, some of these methods have large processing times of up to 2600 seconds [1] and other methods do not achieve reliable detections [2].

We explore a combination of methods that yield reliable copy-move detections in a reasonable amount of time.

Methodology

Our method draws inspiration from a technique for stitching photos into into a panorama [4]. We follow a large portion of the panorama stitching pipeline in that we detect interest points, create oriented descriptors around the interest points, find potential matches, and estimate the transformation from the source to destination. Instead of applying this method to image stitching, we use this technique for identifying key features that have a strong correlation to other portions of the image. This is analogous to "stitching" the photo in question to itself.



Interest points

To get interest points, we found the Harris corners, or points that have a high gradient in both the x and y direction within a 3x3 window [?]. We then use one of two methods to select a subset of the Harris corners: Adaptive Non-Maximal Suppression (ANMS) or gradient-based sorting. In ANMS, ... In gradient-based sorting, we take the Harris corners that have the highest gradient, regardless of their distance to other points.

Descriptors

We experimented with two different descriptors, the box descriptor and SIFT (scale invariant feature transform).

For the box descriptor, we first implemented a naive box descriptor that did not take orientation into account. The box descriptor is created by sampling a 40 x 40 window that is Gaussian blurred (at $\sigma = 1.0$) and then downsampled to an 8 x 8 window, which is subsequently rescaled such that the features are zero mean unit variance. This descriptor worked well in identifying strictly translational copy move instances, but was a poor descriptor for rotated copy move instances.

In order to make our box descriptor rotationally invariant, we oriented the images such that the dominant gradient direction was aligend with the axis of the 40 x 40 window. The dominant gradient was found by bucketing each pixel's orientation into ten 36 degree buckets, which was weighted by the gaussian coefficient based on each pixel's distance to the central interest point and the magnitude of each pixel's gradient. The weighted average of all the point orientations that were a part of the dominant gradient direction was used to then calculate the overall weighted orientation of the descriptor.

For SIFT, for each interest point, we take a 16x16 window around that point, splitting it into 4x4 windows. For each 4x4 window, we calculate the orientation and magnitude of the gradient at that point and bucket them into 8 buckets. We do this for each of the 16 4x4 windows for a total of 128 feature descriptors. Then, we subtract the magnitude and orientation of the interest point and normalize. To avoid any values that are significantly higher than the rest, we threshold the normalized vector at 0.2 and renormalize.

Matching

After we have descriptors, we use the

Want to reduce the number of potential matches (filtering)

Table 1: Evaluation Metrics

	Box, ANMS	Box, High	BR, ANMS	BR, High	SIFT, ANMS	SIFT, High
# Points	0	0	0	0	0	0
Time (s)	0	0	0	0	0	0

Estimated Transformations

We use RANSAC.

Results

We benchmark our results on two metrics: number of points that were correctly matched, and time in seconds for completion. Additionally, we evaluate our results empirically and provide some examples that worked well and others that worked less well. Our numeric metrics are displayed in Table 1.

Discussion and Future Work

Discuss here...

Acknowledgements

Acknowledge here...

References

SAMPLE

1. G. Gamow, *The Constitution of Atomic Nuclei and Radioactivity* (Oxford Univ. Press, New York, 1931).

2. W. Heisenberg and W. Pauli, *Zeitschr. f. Physik* **56**, 1 (1929).