Copy Move Detection

Anthony Sutardja, Kevin Tee

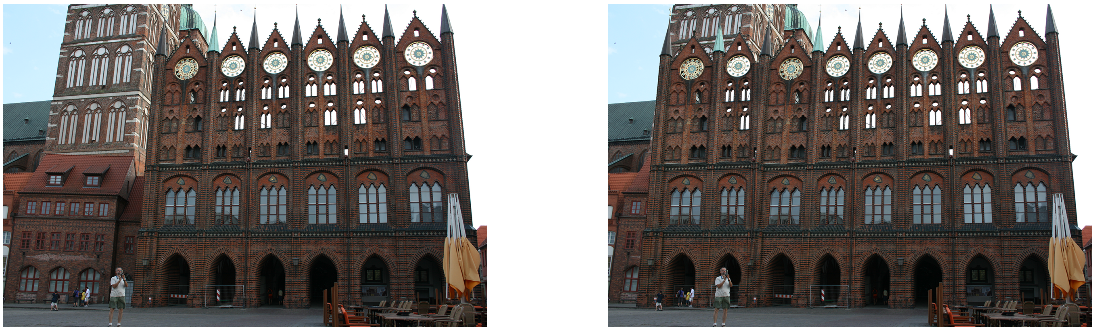
Department of Computer Science
University of California, Berkeley
CS294-26 Final Project

{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 and multi-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.

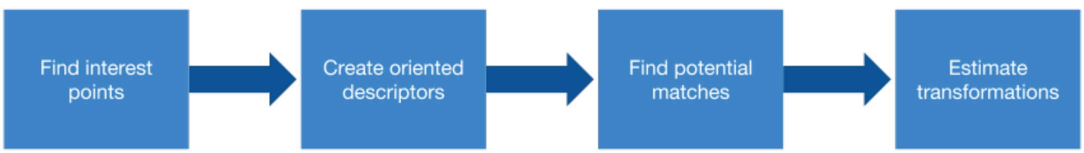
In this paper, we examine the detection of a particular type of image forgery known as copy-move forgery. Copy-move forgery is when portions of the image are resampled to another part of the image with the intent to change the photo's meaning and context. An example of such an attack is replacing a portion of the image with more buildings as seen in Figure 1 [2].

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 (around 45 minutes) [1] and other methods do not achieve reliable detections [7].

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 [6]. 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 within the image that have a strong correlation to other portions of the image. This is analogous to "stitching" the image in question to itself.

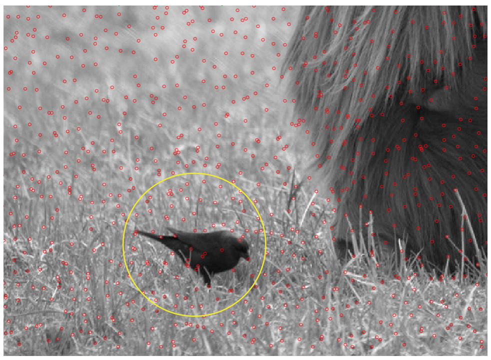


Interest points

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 [6]. We then use one of two methods to select a subset of the Harris corners: corner-based sorting or Adaptive Non-Maximal Suppression (ANMS).

The reason for filtering interest points is for speed. Filtering the points for a subset is necessary to drastically decrease running time. Although using every interest point found can lead to very accurate detections, comparing every interest point is computationally expensive and leads to extremely high processing times [1]. Furthermore, the filtering is also helpful for reducing the number of iterations that RANSAC needs to estimate the various copy-move transformations.

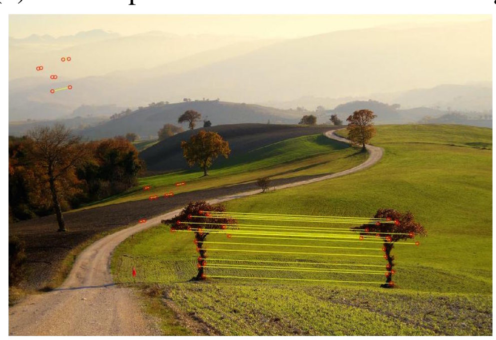
The two different interest point filtering methods both have significantly different use cases.



(a) Interest points from ANMS

(b) Interest points from corner-based sorting

(c) Detection with top harris corners



(d) Detection with ANMS

Figure 2: Figure 2(a) and 2(b) shows the contrast between using ANMS vs corner based sorting on small regions of manipulation; the bird has been copied and corner based sorting is able to capture more interest points for matching. Figure 2(c) and 2(d) show the difference between the locality of detection from corner-based sorting and the spread of detection from ANMS.

In corner-based sorting, we take the Harris corners that have the highest values, regardless of their distance to other points. Taking the top N interest points from corner-based sorting helps capture more localized details. This is especially important with images that have extremely small manipulations, since copy-move transformation estimation requires at least a few points in order to come up with an accurate transformation.

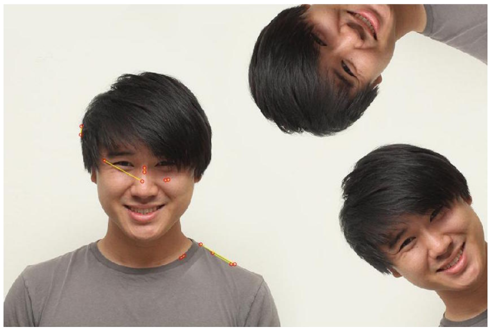
In ANMS, we take Harris corners that are high in value and spread relative. This methodology is directly taken from the MOPS paper; please see the paper for more details on its implementation [6]. Using ANMS helps detection of a broader region of copy-move instances since the collection of interest points are more spread out. This is useful in capturing the full context of the copy-moved manipulation.

Descriptors

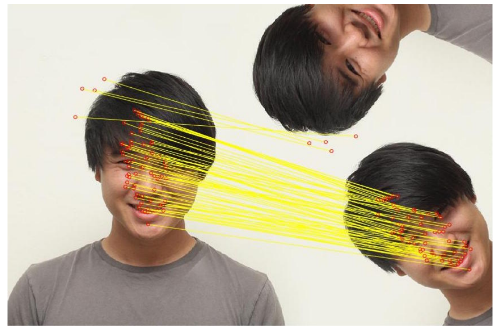
We experimented with two different descriptors, the box descriptor and scale invariant feature transform (SIFT) descriptor.

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 40x40 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 windows such that the dominant gradient direction was aligned with the axis of each Gaussian blurred 40x40 window (with $\sigma = 4.0$). The dominant gradient was found by bucketing each pixel's orientation into ten



(a) Without oriented box descriptors



(b) With oriented box descriptors

Figure 3: The figure shows how copy-move detection is dramatically improved when orientation is taken into account for the descriptors.

36 degree buckets. Each entry into the bucket 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 placed into the largest bucket was used to then calculate the overall weighted orientation of the descriptor. We also included the weighted orientation of buckets that were local maximas (taking the buckets that were at least 0.8 times largest peak). This technique worked very well in identifying rotational copy-move as seen in Figure 3.

In SIFT, for each interest point, we take a 16x16 window around the interes 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 sixteen 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 [1].

Matching

After we have descriptors, we use the outlier rejection with nearest neighbor. Specifically, using a basic distance metric, Euclidean distance (our feature vector/descriptor is relatively low dimensional), to find the first and second nearest neighbor for each descriptor, and then we take the ratio of the distance. If that ratio is less than a threshold value, which we set to be 0.4, which the optimal value for panorama stitching [6]. In our experiments, we also tried to vary the threshold value. A smaller threshold value finds stronger matches, and subsequently makes running the estimated transformations faster. However, it can miss out on points. A large threshold will find more matches, but increases the running time of finding the set of estimated transformations. With our application, 0.4 was a good tradeoff between speed and breadth of points. A small detail worth mentioning is that for each descriptor point, we had to set the distance to itself to be infinity, otherwise it would find itself to be the nearest neighbor.

To improve the speed of the subsequent step, we also do an additional step of filtering. We found that many points that were returned by the matching were points that were a few pixels away from each other, since the texture doesn't change too much. However, it is difficult by hand to copy-move anything less than 10 pixels (or even 15-20 pixels). Therefore, we remove matches that are 3 pixels or less apart.

Estimated Transformations

Matching provides us with many correspondence points that have similar features, and we now use RANSAC to find the best set of matches that correspond to a copy move. RANSAC is an iterative method that repeatedly selects 4 points and computes the sum of squared differences (SSD) of each set of corresponding points [6]. Then, it keeps the set of inliers, where the SSD is less than some epsilon values. We run this for 8000 iterations.

However, RANSAC has a single inlier set and therefore does not scale well to multiple translations. We derive a new method, multi-RANSAC (mRANSAC) that solves this issue. Multi-RANSAC performs exactly the same as RANSAC, except that it saves the top k inlier sets (we used $k = 6$). This essentially attempts to capture multiple transformations, which enables us to identify more than one specific instance of copy-move within an image. In this modified version of RANSAC, we also only kept inlier sets that contained a minimum number of points that was set by a threshold.

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. More examples of successful detections are listed in the appendix.

Discussion and Future Work

Overall, we are satisfied with our copy-move detection algorithm as it has correctly classified most instances in our dataset. Our results outperform most other algorithms in detection time and accuracy. For the majority of the algorithms that we compared our method to, our algorithm was able to identify more copy-move regions with less false positives (we had no false positives). For the single algorithm that performed better in accuracy than our method [1], our method was able to identify a sparser copy-move region in far less time (45 minutes vs 15

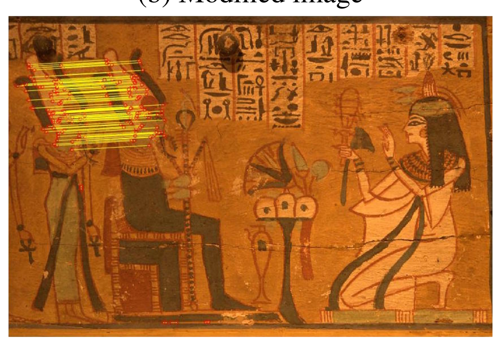
Table 1: Evaluation Metrics for image Egyptian in Figure 4. The table shows the running time as well as the number of matched points for each method on the egyptian copy image. We see that the unoriented box (B) and SIFT descriptors are much faster than the rotationally invariant box descriptor (BR), while achieving roughly the same performance. Using ANMS against corner-based sorting (Sort) resulted in lower runtimes primarily because corner-based sorting captured many more interest points in order to achieve detection.

	B, ANMS	**B, Sort**	**BR, ANMS**	**BR, Sort**	**SIFT, ANMS**	**SIFT, Sort**
# Points	40	93	41	96	15	44
Time (s)	14.67	18.87	115.56	307.41	14.31	16.64

(a) Original image

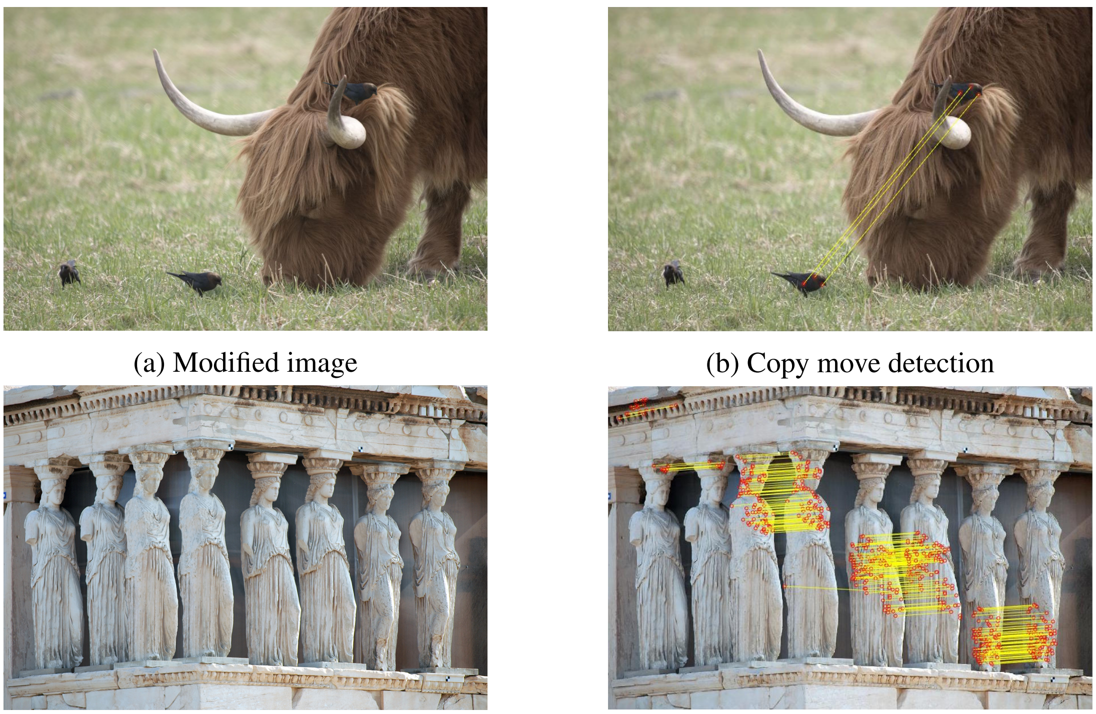
(b) Modified image

(c) No copy move detection



(d) Copy move detection

Figure 4: Figure 4(a) and 4(b) shows the original and the modified image. Figure 4(c) and 4(d) show that our algorithm finds no copy move for the original image, but a copy move for the modified image.



(a) Modified image

(b) Copy move detection

(c) Modified Image

(d) Copy move detection

Figure 5: Figure 5(a) and 5(b) shows the matched points for the cattle image. Our algorithm is able to find these points despite the lack of edges on the bird and the small rotations. Figure 5(c) and 5(d) shows the matched points for the kore image. Our algorithm is able to find multiple sets of copy move due to mRANSAC.

seconds).

There are two key insights from our work on copy-move detection. First, there is a trade-off between speed and accuracy. More interest points can lead to more accurate and complete detection, but at the expense of time. Using a sparse set of interest points decreases running time significantly, but forces us to choose a strategy in selecting the optimal gsparse set of interest points. This leads to our second insight: the two different strategies that we tried have their different strengths. Filtering by ANMS enables our algorithm to capture the overall context of the copy-moved region, while filtering by corner-based sorting enables our algorithm to capture smaller copy-moved patches. Using a combination of both these filtering techniques helps in identifying copy-move forgeries.

One area of future work includes detecting scaled copy move detection. Currently SIFT is not sufficient for detecting copy move, and we propose a method to solve this problem: repeatedly downsize an image and run feature matching. However, this tends to be less common in copy move examples, as objects can appear out of place with too much scaling and are therefore more easily detected by the human eye. Another, albeit much more difficult, problem is to detect points that were warped with some homography. One potential solution to solving these problems is to instead form interest points in the frequency domain, a method that we hope will lead to uncover more underlying structure of an image and portions of the image.

Another area of future work is to formalize mRANSAC and make it more robust to different images and arbitrary number clusters (it's currently only able to capture a pre-defined number of transformations). We propose using some variant of the Hierarchical Dirichlet Process (HDP) to automatically find the clusters and be able to integrate it into the RANSAC algorithm. Our currently methodology is able to find points of the copy move, although it often just finds a portion of the copy move. The criterion of the inlier set could be experimented with in mRANSAC to be able to detect a wider spread of the copy move.

Last, we propose two experiments for future work using our current framework to improve accuracy. The first is finding the optimal window size for each of the descriptors and developing a method of determining what window size to on which images. It is unclear whether using less points will increase or decrease the accuracy of the matching. Increasing points will allow for more matches, but at the same time it will add more “noise”. A formal assessment of this can determine the optimal number of points used.

Acknowledgements

We would like thank Alyosha Efros for teaching and inspiration, as well as the ideas from his panorama stitching project which greatly influenced our work. We thank James O’Brien for his initial guidance and support on this topic. We would additionally like to thank the University of Nurnberg for the dataset [2].

References

1. B. Shivakumar et al. *Automated Forensic Method for Copy-Move Forgery Detection based on Harris Interest Points and SIFT Descriptors*. IJCA, August 2011.

2. D. Lowe, *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

3. *Image Manipulation Dataset*. http://www5.cs.fau.de/research/data/image-manipulation.

4. J. Fridrich, D. Soukal and J. Lukas. *Detection of Copy-Move Forgery in Digital Images*

with. Proc. of DFRWS 2003, Cleveland, OH, USA, August 5-8 2003

5. M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

6. M. Brown and D. Lowe. *Automatic Panoramic Image Stitching using Invariant Features.* International Journal of Computer Vision. 74(1), pages 59-73, 2007.

7. M. Brown, R. Szeliski and S. Winder. *Multi-Image Matching using Multi-Scale Oriented Patches*. International Conference on Computer Vision and Pattern Recognition (CVPR2005) pp. 510-517.

8. S. Pittala. *Copy Move Forgery Detection using HOG Descriptor*.

Appendix

.

Table 2: Evaluation Metrics for All Images

	B, ANMS	**B, Sort**	**BR, ANMS**	**BR, Sort**	**SIFT, ANMS**	**SIFT, Sort**
Egyptian						
# Points	40	93	41	96	15	44
Time (s)	14.67	18.87	115.56	307.41	14.31	16.64
Cattle						
# Points	0	0	0	3	2	4
Time (s)	13.80	15.80	95.53	333.80	14.90	16.76
Anthony						
# Points	0	0	31	77	2	0
Time (s)	10.48	18.37	79.95	209.06	11.88	15.92
Extension						
# Points	116	372	141	442	77	253
Time (s)	15.26	24.80	99.58	283.00	13.06	18.14
Tree						
# Points	14	53	21	55	14	55
Time (s)	12.54	16.54	87.49	259.82	12.08	15.42
Kore						
# Points	81	280	91	323	67	234
Time (s)	16.78	46.52	114.23	334.97	14.70	18.22

Figure 6: The figures above show successful examples of copy-move detection.