

Experiments in Building Recognition

Wei Zhang

wzhang2@cs.gmu.edu

Jana Kořecká

kosecka@cs.gmu.edu*

Technical Report GMU-CS-TR-2004-3

Abstract

In this report we study the problem of building recognition. Given a database of building images, we are interested in classifying a novel view of a building by finding the closest match from the database. We examine the efficacy of local scale-invariant features and their associated descriptors as representations of building appearance and use a simple voting scheme in the recognition phase.

1 Introduction

Object recognition has been an active research area in computer vision for a long time. Various approaches have been proposed to recognize a variety of dissimilar objects coming from different classes. Buildings have several characteristics which make the recognition more challenging. They are highly structured, with parallelism and orthogonality being the prevailing relationships between lines. They typically have repeating structures, such as windows, pillars (see Figure 1). In images of buildings occlusions happen frequently, due to trees around them or self-occlusions due to the change of viewpoint. We consider scenarios where the dominant structure in the image is due to buildings. This differs from related work on detecting and recognizing buildings in aerial images.

In this report we are interested in investigating the efficacy of local image descriptors as building representations, their detection repeatability as a function of viewpoint as well as discrimination capabilities. Given a database of building images and a query image with a novel view of a building, we want to find the closest



Figure 1: Some examples with repeating structures within each image.

match from the database.

2 Related work

The majority of approaches for dealing with classes of objects which share common visual attributes such as cars, faces or buildings focused mostly on the problem of detection and class classification. In [18] authors worked on locating buildings in a given image or classifying images as building/no building images in the context of CBIR application [7]. More challenging problem of detection of man-made structures in cluttered scenes has been addressed in [5], where the authors modelled the image spatial dependencies using Markov Random Fields.

The crucial part of a object recognition system is object representation. Currently existing approaches can be broadly divided into two categories: appearance based and geometric techniques. Geometric techniques model objects using geometric information such as shape [21],

*This work is supported by NSF IIS-0118732

[2], [3], or coordinates of points [20]. Rather weak discrimination capabilities of purely geometric methods as well as difficulties of employing these techniques in the presence of clutter make them mostly applicable in scenarios where detailed geometric models of objects (such as provided by CAD models) are available.

Appearance based techniques represent objects by their appearance, and can be further divided into global or local appearance based methods. Principal Component Analysis (PCA) [19], [12] has been traditionally used in a global setting. The disadvantage of the standard PCA techniques is that they can't tolerate occlusions and clutter. In [6] authors propose to handle occlusion problems of PCA by considering only subset of image windows. Local appearance based methods represent objects by a set of local image descriptors [16] associated with feature points. In [13] authors introduce "eigen-windows" which are also based on eigen-space analysis.

Recently several feature detectors and their associated local descriptors have been proposed in the literature. Mikolajczyk and Schmid [11] present an affine invariant interest points detector. They initialize interest points based on a multi-scale Harris corner detector, and iteratively modify the position, scale and shape of those points. They use normalized Gaussian derivatives as local descriptor, and get point to point correspondences based on Mahalanobis distance between two descriptors. Rothganger [14] extends their work to multi-view case, by exploiting multiview geometric constraints between sets of matched interest points. Spin image is used to represent the normalized image patch and normalized cross-correlation of the descriptors is used for matching. Lowe [10] introduced so-called SIFT features by selecting stable features in the scale space. We describe SIFT features in greater detail in the next section. Nelson and Selinger [17] employ a hierarchy of perceptual grouping process. Their basic primitives are keyed context patches, which are square image region extracted by taking prominent contour fragments, the orientation and size of the regions are normalized by the contour.

To the best of our knowledge, the only work related to building recognition is by Bohm and Haala [4]. They use overall shape of the building to recognize buildings. The requirement is much more than image, including CAD model, GPS receiver and compass.

Approach In this report we evaluate the performance of building recognition using the scale invariant features introduced in [10]. The set of local descriptors for each reference image represent model for each building. In the recognition stage, each descriptor of a query image is matched against the database of models. The number of matches between the pair of query and model images

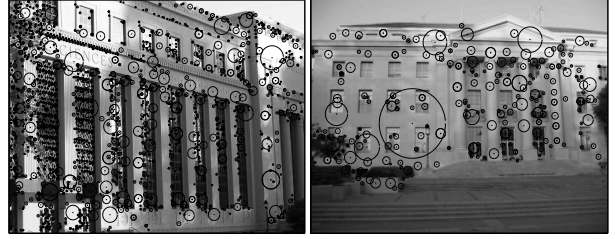


Figure 2: Examples of SIFT features. The size of the circle is proportional to scale of the feature.

is then used to determine the best match in the database. In the following sections we briefly describe the process of selecting the SIFT features, the distance metric used to compare the descriptors and the final recognition stage. The experiment has been carried out on the database of 36 buildings.

3 Building representation

The buildings are represented by a set of scale invariant SIFT features and their associated descriptors. The SIFT features correspond to highly distinguishable image locations which can be detected efficiently and have been shown to be stable across wide variations of viewpoint and scale. Such image locations are detected by searching for peaks in the image $D(x, y, \sigma)$ which is obtained by taking a difference of two neighboring images in the scale space:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (1)$$

The image scale space $L(x, y, \sigma)$ is first built by convolving the image with Gaussian kernel with varying σ , such that at particular σ , $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$. This difference operation approximates the convolution of an image with Laplacian of Gaussian $D(x, y, \sigma) \approx \sigma^2 \nabla^2 G * I(x, y)$, with additional scale normalization [8]. Candidate feature locations are obtained by searching for local maxima and minima of $D(x, y, \sigma)$. In the second stage the detected peaks with low contrast or poor localization are discarded. The location of keypoints can be further refined by fitting a 3D quadratic function to local point neighborhood to get sub-pixel accuracy in [9, 1].

3.1 Keypoint descriptor

Orientation of keypoint is determined by peaks in local orientation histogram, which is formed around a circular window of the keypoint. The window size is 3 times that

of the scale of the keypoint. By assigning a dominant orientation to each keypoint, the keypoint descriptor is represented relative to this orientation in order to achieve the invariance to image rotation.

After assigning scale and orientation to each keypoint, a descriptor is created characterizing a local image area around each keypoint: First, the gradient magnitude and orientation are computed at each pixel. The local neighborhood of each keypoint is then represented by 16 local orientation histograms, each belonging to one of the 4×4 sub-windows of the local keypoint neighborhood with 8 orientation bins in each. This forms a 128 (16×8) element feature vector for each keypoint. The boundary effects between histograms are alleviated by distributing gradient value of each pixel to adjacent histogram bins via linear interpolation. Although the descriptor is not fully affine invariant, it has been shown to be robust across a substantial range of affine distortions [10]. Examples of buildings and their associated scale invariant features are in Figure 2. Given a database of reference views of buildings each is characterized by a set of keypoints and their corresponding descriptors. The descriptors are then normalized in order to tolerate moderate changes in image contrast. The normalized descriptors are saved in the model database.

3.2 Image Matching

Given a query image and model image, the matching between the two is based on two criteria. As proposed by Lowe, we first for each keypoint descriptor $d \in \mathbb{R}^n$ find the two closest neighbors. As long as the ratio of distances between the closest and second-closest keypoint descriptor is below certain threshold the match has been successful. The lower the ratio, the better the match. If the ratio is below some threshold τ_r , the point corresponding to the descriptor is labelled as matched. In practice, we use the squared distance instead of Euclidean distance so a point is labelled as matched when

$$\frac{d^2(\mathbf{f}, \mathbf{m}_{1st})}{d^2(\mathbf{f}, \mathbf{m}_{2nd})} < \tau_r^2 \quad (2)$$

where \mathbf{f} is the descriptor to be matched and \mathbf{m}_{1st} and $\mathbf{m}_{2nd} \in \mathbb{R}^n$ are the closest and the second closest descriptors from the model database. In our case $n = 128$. In [9] Lowe suggests using $\tau_r = 0.8$. Generally, this measure is effective because correct matches need to have the closest neighbor significantly closer than the closest incorrect match. In our case, it is not always true, because there are many repeating patterns in buildings, such as corners of windows. Consequently it is very likely that a keypoint has several close matches with the distance ratio between the first and second neighbors being close to 1. In order to retain these matches we use

additional criterion related to the actual distance between the descriptors. If the cosine of the angle between two descriptors is above certain threshold τ_c , we also label them as matched. Given two descriptors a and b their cosine distance is defined as:

$$\cos(a, b) = \frac{a^T b}{\|a\|_2 \|b\|_2} \quad (3)$$

The cosine measure reflects the angle between two vectors (descriptors). The square of Euclidean distance between two descriptors can be obtained directly from their cosine value as:

$$\begin{aligned} d^2(a, b) &= \sum_i a_i^2 + \sum_i b_i^2 - 2 \sum_i a_i b_i \\ &= \|a\|^2 + \|b\|^2 - 2 \cos(a, b) \|a\| \|b\| = 2(1 - \cos(a, b)) \end{aligned}$$

The using the second criterion we can retrieve larger number of correct matches, which would otherwise be rejected by the first criterion. Since we don't want to introduce many false matches the threshold τ_c is rather high.

3.3 Threshold selection

The choice of thresholds can significantly affect the recognition rate. Ideally we would like to choose such thresholds which would guarantee large number of correct classifications while keeping the number of false positive matches low. The choice of suitable thresholds is typically based in ROC (Receiver Operating Curve) analysis. ROC curve capture the relationship between the *sensitivity* (True Positive Rate) and the *specificity* (True Negative Rate). Its y-axis is the True Positive Rate (TPR), and x-axis is False Positive Rate (FPR) which is exactly 1-Specificity. The TPR are defined as follows:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \quad (4)$$

where TP represents the number of true positives (in our case the number of correct matches found given a threshold), P represents the total number of positive (correct matches), and FN represents the number of false negatives (i.e. the number of correct matches that are not found given a threshold). The FPR is defined as

$$FPR = \frac{FP}{N} = \frac{FP}{TN + FP} \quad (5)$$

where FP represents the number of false positives (in our case, the number of incorrect matches found given a threshold), N represents the total number of negatives (incorrect matches), and TN represents the number of true negatives (i.e. the number of incorrect matches that are not found given a threshold).

The values of the individual quantities are rather difficult to obtain in the context of our application. First, numbers P and N are hard to determine. Given the fact that number of keypoints in each image is around 800, even for one image pair counting the number of correct matches is nontrivial. The situation becomes worse because of the repeating structure which causes the keypoint to "correctly" match to multiple positions. Furthermore in order to study how the thresholds affect the performance, we need to identify FP and TP for match results returned by a range of thresholds. These in our application can be determined only tediously by hand. In order to determine the appropriate value of the thresholds we approximate the ROC curve as follows. We randomly choose one test image, and match it's keypoints to all the keypoints in the 36 models. For each model, a set of matched keypoints is returned. All the matched pairs except those from the correct model are false positives (FP). For matched pairs from the correct model, we still need identify the true positives (TP), because they can include wrong matches. Once we know the FP and TP for the chosen threshold, we need to obtain N and P . In fact we are really interested in the ratio of N and P . Once their ratio is set, the shape of the ROC curve wouldn't change with regard to different N and P . Let's consider P first, in order for the set of match pairs to include all the true positive, the threshold should be minimal strict. i.e. the similarity threshold should be 0, while the ratio threshold should 1. Of course, false positives are also included in the set, our observation shows that around 2/3 of matched pairs are true positives. Matched pairs produced from all the remaining 35 models are false positives. Let the number of keypoints in the test image be N_q , the $N \approx (35 + 1/3)N_q$ and $P \approx 2/3N_q$. The ratio N/P is approximately 50. After all those parameters are set, we can draw the ROC curve. Of course, one test image is not enough to represent the variation of whole dataset. In this experiment we use four test images to study the behavior of the threshold, and use their average to draw the approximate ROC curve. See Figure 3 and 4 for final result. The threshold we choose are $\tau_c = 0.97$ and $\tau_r = 0.6$.

3.4 Keypoint matching

Each keypoint is labeled as matched if either of the two criterias is met, thus we get more correct matches. Figure 5 shows the benefit of combining the Lowe's measure and cosine measure.

Because of repeating patterns, each particular keypoint may not be matched to its exact position, as Figure 7 shows. For example a window corner may be matched to a similar window corner with completely different location. The match is correct in the meaning of

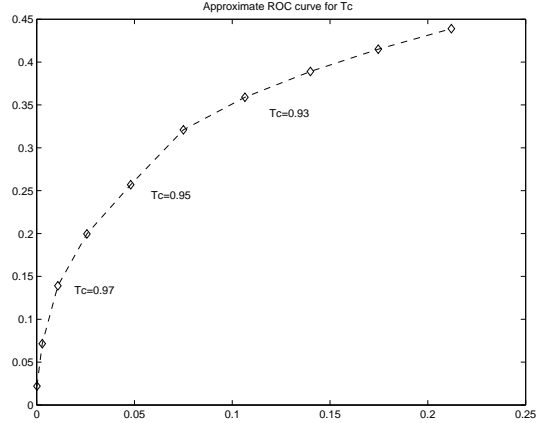


Figure 3: Approximate ROC curve for τ_c .

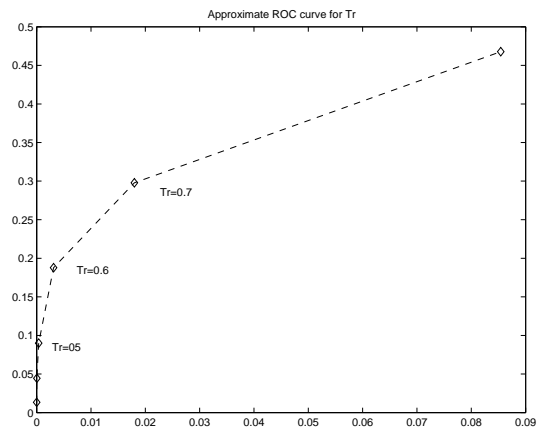


Figure 4: Approximate ROC curve for τ_r .

local appearance matching, but the points are not in correct correspondence in the sense that they are not projections of same 3D point. Fortunately, this is not a big problem for recognition purpose, because we only need to know whether the point come from the building.

4 Building recognition

The building recognition is accomplished here by a simple voting scheme. Given a query image and sets of matched keypoints pairs, with all existing models in the database, the most likely model is the one with the largest number of matches.

Although the geometric consistency between matched keypoints can further improve the recognition result, the complexity involved in achieving it hinders us from pursuing it. The robust technique like RANSAC is known to handle 50% outliers, in our case, there are often well above 50% outliers. Mismatches constitute part of outliers, furthermore, repeating patterns cause many locally



Figure 5: (left) Match result using only Lowe's criteria (15 matches); (center) result using only cosine measure (9 matches); (right) result using both measure(22 matches).

correct yet globally mismatched keypoint. In the extreme case, all the matched keypoints may belong to the later category. Consequently, it's hard to define geometric consistence, yet recognition is still successful. The examples in Figure 7 show some extreme situations.

5 Experimental results

The image database used in the experiment consists of 36 buildings, with several images for each building. The images are taken from different viewpoints and have different scales. We use one image as reference image and two other images for testing. All of the buildings are listed in Tables 6, 7 and 8.

In the experiment 53 out of 72 test images are correctly recognized (listed as first candidate), the rest of them are misclassified. But 10 out of them have correct answer listed in top 5 list, so the results are still useful for further processing. The detailed results are listed in Table 1 and Table 2. For the test images which get completely wrong results (the correct answers are not shown in the top 5 list), possible reasons are either too large out-of-plane rotation (e.g. image 25) or scale change (e.g. image 14). In addition, the lighting conditions maybe rather different between reference and test images. We are currently in the process of extending our database and carrying out additional experiments. We plan to further investigate the alternative choice of features, their descriptors as well as

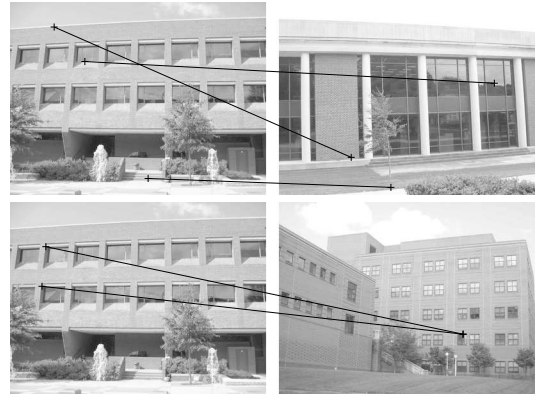


Figure 6: Keypoints match results to incorrect models.

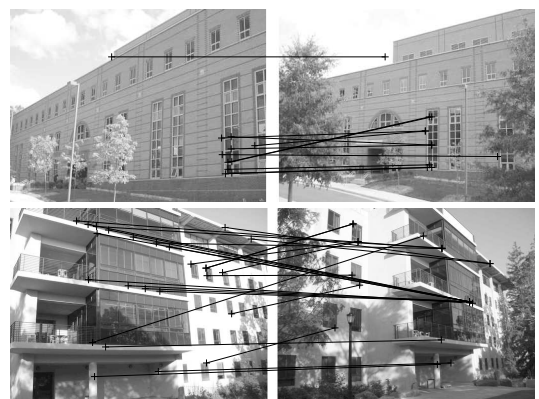


Figure 7: Repeating structures cause mismatches, although in term of local appearance, they are correct.

means of modelling spatial relationships between the local keypoints.

6 Conclusion and future work

In this report we have presented initial experiments in building recognition using scale invariant features and their associated descriptors. We examined two different criteria for matching and demonstrated that their combination yields favorable results in the context of our domain. Currently we can only deal with single building in the image and do not exploit spatial relationships between buildings. We are currently extending the database of features and developing the additional stage which would enable robustly estimate the pose of the camera with respect to the building.

References

- [1] M. Brown and D. Lowe. Invariant features from interest point groups. In *In Proceedings of BMVC, Cardiff, Wales*, pages 656–665, 2002.
- [2] A. Dubinskiy and S. Zhu. A multiscale generative model for animate shapes and parts. In *International Conference on Computer Vision*, October 2003.
- [3] B. Falcidieno and F. Giannini. Automatic recognition and representation of shape-based features in a geometric modeling system. *Computer Vision, Graphics, and Image Processing*, 48:93–123, 1989.
- [4] P. K. Jan Bohm, N. Haala. Automated appearance-based building detection in terrestrial images. In *International Archives on Photogrammetry and Remote Sensing IAPRS*, pages 491–495, September 2002.
- [5] S. Kumar and M. Hebert. Man-made structure detection in natural images using a causal multiscale random field. In *CVPR 2003*, pages 119–126, 2003.
- [6] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pages 453–458, June 1996.
- [7] Y. Li and L. G. Shapiro. Consistent line clusters for building recognition in cbir. In *CVPR 2002*, pages 952–956, August 2002.
- [8] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2), 1994.
- [9] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, page to appear, 2004.
- [10] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV*, pages 1150–1157, 1999.
- [11] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142, 2002.
- [12] H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.
- [13] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1043–1047, 1997.
- [14] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *CVPR 2003*, pages 272–277, June 2003.
- [15] R. Schettini. Multicolored object recognition and location. *Pattern Recognition Letters*, 15(11):1089–1097, November 1994.
- [16] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence*, 19:530–535, 1997.
- [17] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance-based 3D object recognition. *Computer Vision and Image Understanding: CVIU*, 76(1):83–92, 1999.
- [18] A. Stassopoulou. Building detection using bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 83(5):705–740, 2000.
- [19] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3:71–86, 1994.
- [20] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings of the third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 454–459, April 1998.
- [21] S. C. Zhu and A. Yuille. Forms: a flexible objects recognition and modeling system. *Computer Vision, Graphics, and Image Processing*, 20:187–212, 1996.

	First	Second	Third	Fourth	Fifth	Correct?		First	Second	Third	Fourth	Fifth	Correct?
1	1	29	5	26	11	1	1	1	29	2	5	20	1
2	2	9	29	1	5	1	2	2	29	5	4	13	1
3	3	29	1	23	11	1	3	3	29	2	11	34	1
4	4	5	8	2	29	1	4	4	5	26	28	2	1
5	5	7	6	12	24	1	5	2	5	29	4	20	0
6	6	5	11	29	1	1	6	6	2	29	26	34	1
7	7	2	8	5	28	1	7	7	8	26	28	29	1
8	8	29	28	17	24	1	8	8	29	2	6	1	1
9	5	9	29	2	28	0	9	2	29	11	34	1	0
10	10	29	9	4	5	1	10	10	11	20	29	27	1
11	11	5	1	29	4	1	11	11	1	29	5	20	1
12	5	12	28	6	9	0	12	5	20	30	2	29	0
13	13	9	4	29	5	1	13	13	2	23	29	9	1
14	14	1	29	2	9	1	14	1	29	11	23	34	0
15	15	5	8	28	2	1	15	29	1	15	5	28	0
16	16	5	29	9	2	1	16	29	20	16	23	11	0
17	17	15	5	29	11	1	17	29	11	23	6	18	0
18	18	5	11	6	28	1	18	2	11	27	26	20	0
19	19	5	1	3	23	1	19	19	34	26	5	23	1
20	20	5	8	11	1	1	20	20	35	29	5	28	1
21	21	7	2	29	6	1	21	29	4	35	28	2	0
22	22	5	28	2	29	1	22	22	35	2	31	9	1
23	23	29	2	5	6	1	23	23	29	5	26	35	1
24	29	28	24	5	2	0	24	35	29	28	2	24	0
25	2	5	29	7	1	0	25	2	25	12	7	29	0
26	29	26	28	5	11	0	26	26	35	2	11	29	1
27	27	5	2	29	23	1	27	27	5	29	28	2	1
28	28	29	5	1	9	1	28	28	35	29	6	2	1
29	29	2	1	4	5	1	29	29	35	2	23	28	1
30	30	5	8	14	20	1	30	30	29	5	28	2	1
31	31	11	1	5	4	1	31	31	35	5	23	1	1
32	5	20	26	32	1	0	32	32	35	5	34	26	1
33	34	1	26	35	28	0	33	35	29	1	2	9	0
34	34	26	35	11	1	1	34	35	34	29	2	26	0
35	35	34	26	17	29	1	35	35	29	2	5	28	1
36	36	35	34	29	5	1	36	36	5	28	35	29	1

Table 1: Recognition results with top 5 list for first test image

Table 2: Recognition results with top 5 list for second test image




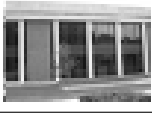
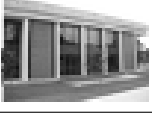
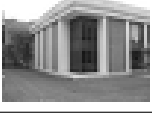



























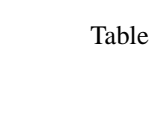
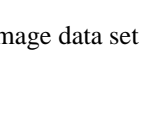

	Reference image	Test image	
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Table 3: Image data set


































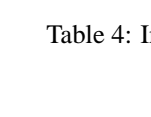
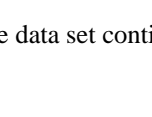

	Reference image	Test image	
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			

Table 4: Image data set continue



















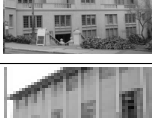



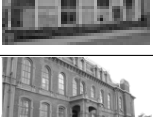










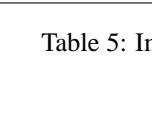
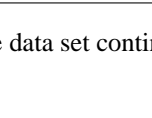

	Reference image	Test image	
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			

Table 5: Image data set continue