Open access • Proceedings Article • DOI:10.1109/ICCV.2011.6126508

Evaluation of image features using a photorealistic virtual world — Source link

Biliana K. Kaneva, Antonio Torralba, William T. Freeman

Institutions: Massachusetts Institute of Technology

Published on: 06 Nov 2011 - International Conference on Computer Vision

Topics: Image-based lighting, Feature (computer vision), Feature detection (computer vision), Image-based modeling and rendering and Image processing

Evaluation of Image Features Using a Photorealistic Virtual World

Biliana Kaneva biliana@csail.mit.edu
Antonio Torralba torralba@csail.mit.edu
William T. Freeman billf@mit.edu
MIT Computer Science and Artificial Intelligence Laboratory

Abstract

Image features are widely used in computer vision applications. They need to be robust to scene changes and image transformations. Designing and comparing feature descriptors requires the ability to evaluate their performance with respect to those transformations. We want to know how robust the descriptors are to changes in the lighting, scene, or viewing conditions. For this, we need ground truth data of different scenes viewed under different camera or lighting conditions in a controlled way. Such data is very difficult to gather in a real-world setting.

We propose using a photorealistic virtual world to gain complete and repeatable control of the environment in order to evaluate image features. We calibrate our virtual world evaluations by comparing against feature rankings made from photographic data of the same subject matter (the Statue of Liberty). We find very similar feature rankings between the two datasets. We then use our virtual world to study the effects on descriptor performance of controlled changes in viewpoint and illumination. We also study the effect of augmenting the descriptors with depth information to improve performance.

1. Introduction

Image features play an important role in computer vision. They are used for tasks ranging from wide baseline stereo matching [21, 16, 23], panorama stitching [1] and 3D scene reconstruction [27] to object [2, 4, 5, 10], scene [9, 14, 28], texture [8] and gesture recognition [6]. Parikh and Zitnick [15] studied human performance in visual recognition tasks compared to that of a state-of-the-art computer vision algorithm, and found that, under the conditions of the study, the humans' better performance could be attributed to their better use and selection of image features. Because of their importance and wide use, optimizing image features is a critical task.

The goal in designing features is that they must be robust, distinctive and invariant to various image and scene transformations. One of the challenges is acquiring ground truth data necessary for evaluating and comparing different image descriptors. Mikolajczyk et al. presented a dataset of several images under various transformations [12, 18] addressing this need. Due to difficulty of attaining correspondences, the dataset was limited to planar scenes or images taken from a fixed camera position. These do not capture the full complexity of viewpoint changes - changes in perspective beyond those of planar scenes or the presence of occlusions. The dataset includes an example of change in illumination simulated by changing the camera settings, essentially changes in brightness and contrast. However, these do not capture changes in light source position that result in shadows and non-uniform changes in intensity.

To address such problems, Winder et al. recently proposed using a data set of patches from several famous landmarks [25, 26]. They used camera calibration and multi-view stereo data of 1000 images for each landmark to find corresponding interest points between the images using estimated dense surface models. While these datasets contain image patches taken from different points of view and under different illumination, it is difficult to evaluate the effect each of these has on the descriptor performance, since the variations in viewpoint, illumination and camera type are uncontrolled. Moreels et al. proposed a dataset of 100 real 3D objects viewed from 144 calibrated viewpoints under three different lighting conditions [13]. However, those do not contain complex scenes and interactions between objects such as occlusions, cast shadows, and inter-reflections. We want to be able to capture a wide range of scenes under different transformations. To gain complete, repeatable control over specific aspects of the environment, we propose using a photorealistic virtual world.

With the great progress in the field of computer graphics in the last two decades, it is possible to generate high quality realistic scenes. Recent work has shown that the use of synthetic image/video data can be used to evaluate the performance of tracking and surveillance algorithms [20], to train classifiers for pedestrian detection [11] and to learn locations for grasping novel objects [17]. We propose the use of highly photorealistic virtual world for the evaluation

Figure 1. Sample images from the virtual world. **Top row**: Virtual City. **Bottom row**: Statue of Liberty.

and design of image features. We generated two data sets of images taken under different illumination and from different viewpoints from high resolution 3D graphics models of a virtual city and of the Statue of Liberty. The images were rendered with 3ds Max's Mental Ray renderer using advanced materials, including glossy and reflective surfaces, high resolution textures, and the state-of-the-art Daylight System for illumination of the scene.

We first seek to calibrate our virtual world evaluations against feature rankings made using photographic data. To control for image content, we compare the performance of feature descriptors on datasets based on real and synthetic images of the Statue of Liberty, and we find very similar feature rankings from the photorealistic and photographic datasets. We then exploit the flexibility of our virtual world to make controlled evaluations that are very difficult to make from photographs. We use our controlled environment to evaluate the effects of changes in viewpoint and illumination on the performance of different feature descriptors. We can also study the effect of augmenting the descriptors with depth information to improve performance.

2. Photorealistic Virtual World Dataset

Fig. 1 shows sample images rendered from the Virtual City and from our calibration scene, the Statue of Liberty.

2.1. Photorealistic City Model

For our virtual city dataset, we used a high resolution city model from Turbosquid [22] containing over 25 million polygons. The model has 12 city blocks with 82 unique buildings with highly detailed geometry and advanced textures from residential and commercial ones to churches, schools, theaters and museums. It also includes parks, sport fields, parking lots, and objects found in a city environment, from lamppost and trashcans to benches and bus stops (although no people). We also added 25 different high resolution vehicles to the model that contain advanced glossy and reflective surfaces. To increase the number of vehicles, we varied their colors. The dataset was rendered using 3ds Max's Mental Ray renderer to produce high quality photorealistic city images.

To light the environment, we used 3ds Max's Daylight system that positions the sun light source automatically after specifying the location, the date and time. We rendered five images for each scene taken at 9am, 11am, 1pm, 3pm and 5pm on a sunny summer August day (Fig. 2 top row). We used a 35 mm camera lens. To automatically render the different scenes, we created a fly-through camera animation simulating a person walking along the city streets and varied the illumination at each camera position. At each camera location, we took three different shots panning the camera at 22.5 degree steps (Fig. 2 bottom row a)). Neighboring locations were close enough to capture the scene at the current camera position from a different viewpoint, e.g. figure 2 bottom row b) shows different viewpoints of the scene captured in the center image of figure 2 bottom row a). In this work, we used 3000 images from 200 different camera locations over several city blocks with 15 images taken at each location - three different camera orientations and five different illumination settings for each orientation. The images were rendered at resolution of 640x480 pixels. No noise or compression artifacts have been added to the images though they can be easily added as postprocessing step. The impact of these phenomena on the performance of image descriptors were studied previously in [12].

2.2. Statue of Liberty

Since the photographic subject can influence feature performance, to study whether our photorealistic virtual world would be a good predictor for descriptor performance in the real world, we compared descriptor performance on a synthetically generated dataset of the Statue of Liberty to that on the real world Liberty dataset of [26]. We purchased a high resolution 3D model of the Statue of Liberty and rendered 625 images at 640x480 resolution. We simulated the camera moving around the statue on the ground level in a circular arc centered at the statue. We rendered the scene at every 10 degrees for 250 degrees around the front of the statue and under five different locations of the sun, simi-

Figure 2. Sample images from the virtual city. **Top row**: Images from a static camera of a scene under different illumination (5 different times of the day). **Bottom row**: a) Scene from a panning camera at 22.5 degree rotation stops. b) Images taken from a different camera viewpoint and location of the center image in a).

lar to our city dataset. We used 4 different camera lenses - 50mm, 85mm, 135mm, and 200mm - to acquire both distant and close up shots. We used the 135mm lens at two different angles - viewing the top and the base of the statue.

3. Feature Descriptors

We used our dataset to evaluate the performance of a selection of commonly-used feature descriptors.

3.1. Scale Invariant Feature Transform (SIFT)

SIFT has been widely used in a variety of computer vision applications from object recognition to panorama stitching. We compute the descriptor similarly to [10]. After initial pre-smoothing of the image by $\sigma = 1.8$, we quantize the gradient orientation at each sample into d directions and bin them in 4x4 spatial grid. Each gradient direction is weighted bilinearly according to its distance to the bin centers. The final descriptor is normalized using a threshold of 0.2 as in SIFT [10]. We used 4, 8 and 16 gradient directions thus creating three descriptors of dimension 64, 128, and 256 - these are referred to as T1a-S1-16, T1b-S1-16, and T1c-S1-16 in [25]. The descriptor was computed over a patch of 61x61 pixels centered at the sample.

3.2. Gradient Location and Orientation Histogram (GLOH)

GLOH was proposed as an extension to SIFT to improve robustness and distinctiveness of the descriptor [12]. We quantized the gradient orientations as in SIFT and then bin them in a log-polar histogram of 3 radial and 8 angular directions. Only the outer bins are divided into 8 directions, thus there are total of 17 bins. The size of the patch around the sample was 61x61 pixels and the final descriptor was normalized similarly to SIFT. We used 4, 8 and 16 gradient directions resulting in 68, 136 and 272 dimensional feature vectors - these are similar to T1a-S2-17, T1b-S2-17, and T1c-S2-17 in [25]. Note that we do not reduce the size of the descriptors in our experiments, unlike [12].

3.3. DAISY

The DAISY descriptors is inspired by SIFT and GLOH, but designed for efficient computation [21]. Learning the best DAISY configuration was proposed by [26]. We compute d gradient orientation maps and then convolve them with different Gaussian kernels depending on their distance from the center. The descriptor is then computed over a log-polar arrangement similar to GLOH. The vectors in each pooling region are normalized before concatenated in the final descriptor. We used three radial and eight angular directions for a total of 25 sample centers including the one at the center of the grid. The image patch is 61x61 pixels centered around the sample. We used 4, 8, and 16 gradient directions resulting in 100, 200, and 400 dimensional feature vectors - these are referred to as T1a-S4-25, T1b-S4-25, and T1c-S4-25 in [25].

3.4. Histograms of oriented gradients (HOG)

The HOG descriptor [2] and its variants [3] have demonstrated excellent performance for object and human detection. Similar to the SIFT [10], the HOG descriptor measures histograms of image gradient orientations but normalizes the descriptor with respect to neighboring cells. We

Descriptor	HOG8	SIFT16	GLOH8	DAISY16
Notre Dame Real	0.898l	0.958	0.961	0.964
Liberty Real	0.885	0.947	0.950	0.953
Liberty Synthetic	0.896	0.950	0.955	0.959

Table 1. Area under the ROC curve for different descriptors on the real Notre Dame and Liberty and the synthetic Liberty datasets. Note the feature rankings on both the real and synthetic datasets is the same despite the variation in individual performance. The feature ranking is the same even across datasets with different image content.

use the same approach as described in [3]. However, we compute the descriptor for 4, 8, and 16 gradient orientation. We only use the descriptor for the cell centered at the sample resulting in very low dimensional feature vectors of 10, 16, and 28 dimensions. The descriptor was computed over a patch of 61x61 pixels covering a neighborhood of 3x3 cells.

3.5. The self-similarity descriptor (SSIM)

The self-similarity descriptor [19] has been shown to perform well on matching objects of similar shape but vastly different local appearance. The idea is to represent the appearance in a local image area around a particular image patch by the "correlation map" of the patch with its neighborhood. The descriptor captures the local pattern of self-similarity. Each descriptor is obtained by computing the correlation map of a 5x5 patch in a window with radius equal to 30 pixels, then quantizing it using a log-polar histogram as in GLOH. We used 3 radial bins and either 8 or 16 angular bins, resulting in 24 or 48 dimensional feature vectors.

4. Evaluation

Keypoints are the image locations where we compute descriptors. We computed keypoints using one of three different methods: spatial local maxima of a Difference of Gaussian (DoG) filter [10], the Harris corner detector [7], and a dense spatial grid at 5 pixel offset. We use the implementation of the keypoint detectors by [24]. For the experiments presented here, we use the DoG keypoints. Since our dataset is synthetically generated, we know the complete geometry of the scene and therefore the pixel correspondences across images. Figure 4 a) shows a pair of images taken from different viewpoints and under different illumination. The overlapping part of the scene and the points in the images for which we have correspondences are shown in figure 4 b). Note that we do not match points in the sky for images from a different viewpoint since we do not have actual 3D coordinates for them. They may, however, be considered in experiments where the camera is static. For each image pair A and B, we compute the descriptors at

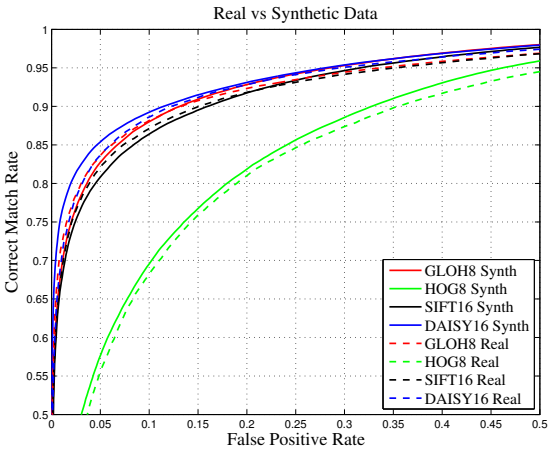

Figure 3. Performance of the synthetic vs real world Statue of Liberty datasets on a set of descriptors. Note that the performance on both datasets is very similar and the relative ranking of the descriptors is the same.

each keypoint in image A and its corresponding 3D point in image B. We define the matching keypoints to be the true correspondences and the non-matching keypoints to be keypoints that are at a distance of at least 10 pixels from the true correspondence in image space. We follow the protocol of Winder et al. [25] to form an ROC curve of descriptor performance. We compute the Euclidean distance between the descriptors computed at each pair of matching and (randomly selected) non-matching keypoints. As a function of a distance threshold, we compute the number of correct and false matches that is the matching and non-matching keypoints with a descriptor distance below the threshold, respectively. Sweeping that computation over a descriptor distance threshold yields a receiver operating characteristic (ROC) curve. The correct match rate and the false positive rate for each discrimination threshold are:

$$\text{Correct Match Rate} = \frac{\#\textit{correct matches}}{\#\textit{matching keypoints}}$$

$$\text{False Positive Rate} = \frac{\#\textit{false matches}}{\#\textit{non-matching keypoints}}$$

The larger the area under the ROC curve, the better the performance of the descriptor.

5. Experiments

5.1. Overview

To first confirm that our virtual world and the real world gave similar rankings, controlling for image content, we compare feature descriptors using the photographic Liberty patch dataset of [26] and our synthetic Statue of Liberty dataset. We find that the descriptors perform comparably on both datasets and the relative rank is the same. We pro-

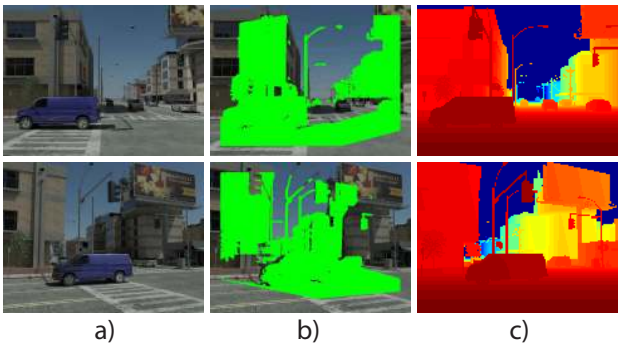

Figure 4. Examples of images from our virtual city. a) Image pair of a scene under different viewpoint and illumination. b) The set of corresponding 3D points between the images in a). c) The corresponding depth maps of the images in a).

ceed to study the effect of changes in illumination of outdoor scenes and changes in camera viewpoint on the descriptor performance. Since our dataset is synthetically generated, we have full control of the scene and we can capture the exact same scene both under different illumination and different camera viewpoint and we have full knowledge of the geometry of the scene that allows to match keypoints accurately. We compare the degradation of all of the descriptors with changes in illumination and viewpoint. We find that the log-polar pooling scheme seems to perform better than the grid one for coping with changes in illumination, while the number of pooling regions has a bigger effect when there are changes in camera viewpoint. We also propose a 3D descriptor in the presence of depth map data and show that even a very low dimensional descriptor like HOG computed over the depth map can lead to improved feature matching performance.

5.2. Real vs Synthetic Data

To calibrate our virtual world descriptor evaluations, we compared the performance on the Liberty patch dataset of [26] and our synthetic Statue of Liberty dataset, using 100000 patches/keypoints in both cases.

For this experiment, we only used images that have a partial or full view of the front of the statue as this seems to be the case for most of the images found online. Figure 3 a) shows performance of a set of the image descriptors on both the real and synthetic data. The ROC curves are very similar showing only slight variation and the ranking of the performance of the different descriptors is the same. The slightly worse performance of the descriptors on the real dataset could be due to inaccuracies in the patch matching. There can be some variation of the descriptor performance depending on the data they are applied to as shown in table 1. To study the change in feature rankings with image content, we kept the evaluation method fixed (photographic image patches) but compared the performance of features

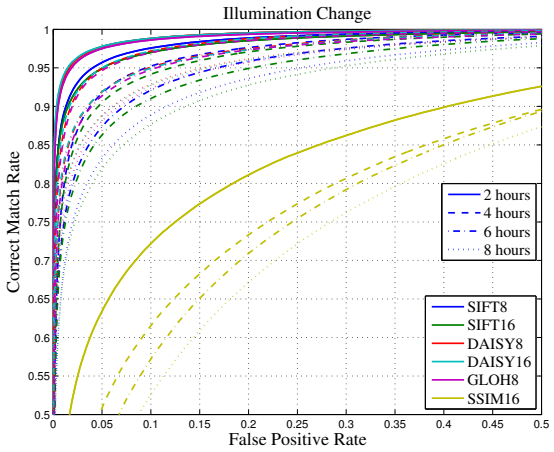

Figure 5. Descriptor performance for images from the virtual city taken with a static camera of a scene under different illumination (2,4,6 and 8 hour difference). The performance degrades with larger changes in illumination. DAISY8 and GLOH8 perform best in this context.

for the Notre Dame dataset [26]. The descriptors perform better on the Notre Dame dataset than on the Liberty one; however, even in this case the ranking of the descriptors is still the same. The better performance on the Notre Dame data set is probably due to the larger number of edge structures in the scene. These results show that (1) we can translate the relative performance of the descriptors on the synthetic data to that of the real data, and (2) the relative rankings appear to change very little across image content.

5.3. Illumination Change

Changes in illumination can result in large changes in the appearance of the scene due to shadows, specular reflections, etc. We compared the performance of the different descriptors under different illumination using our virtual city dataset. Each pair of matching keypoints belonged to images of the same scene taken with a static camera during two different times of day. We used 2.2 million keypoint pairs. Figure 5 shows the performance of a subset of the descriptors for the same scene taken at 2, 4, 6, and 8 hour difference. The performance degrades with the increase of the time difference between the rendered images as the changes in illumination of the scene are more significant. The performance of the other descriptors followed a similar trend. The much worse performance of the SSIM descriptor is likely due to its smaller dimension and lack of distinctiveness as it was meant to be computed densely. The almost identical performance of the DAISY8 and DAISY16 descriptors shows that increasing the number of gradient orientation to 16 is not beneficial. In the case of SIFT, the performance even appears to degrade slightly. DAISY8 and GLOH8 perform very similarly to each other and better than SIFT in the presence of changes in illumination. That may

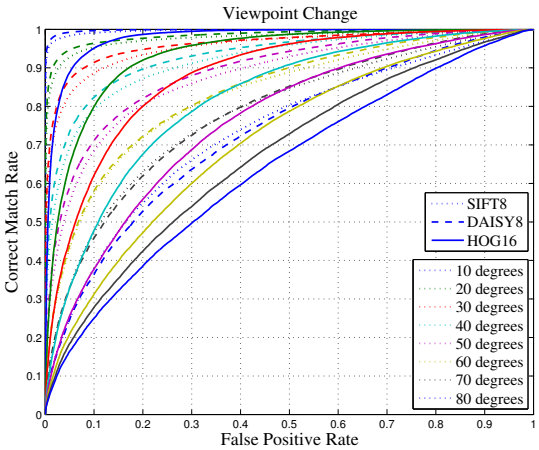

Figure 6. Performance of descriptors on the virtual Statue of Liberty dataset for varying camera viewpoints (10-80 degrees rotation around the statue) under constant illumination. The performance of all descriptors degrades with larger changes in viewpoint. DAISY8 performs better under small changes in viewpoint while SIFT8 performs better under larger changes.

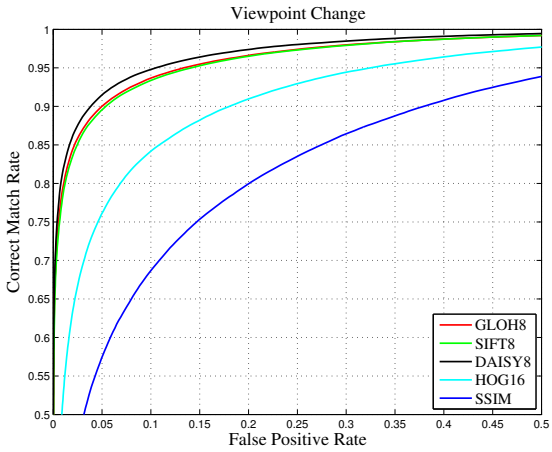

Figure 7. Performance of descriptors under different viewpoint for scenes from the virtual city under constant illumination. Note here GLOH8 and SIFT8 perform similarly, where as GLOH8 performed better than SIFT8 under changes in illumination.

be due to their use of the log-polar binning arrangement, common to DAISY8 and GLOH8.

5.4. Viewpoint Change

We performed a couple of experiments to evaluate the effects of viewpoint change on the different descriptors on both of our datasets - Statue of Liberty and Virtual City.

Our synthetic dataset of the Statue of Liberty contains images taken by moving the camera along a circle around the statue at 10 degree stops. We evaluated the performance of the descriptors as we move the camera up to 80 degrees from the reference image on images taken under the same lighting conditions. Figure 6 shows the performance of several descriptors and how it degrades with the increase in angle between the camera locations. The performance of the DAISY8 descriptor degrades faster after 50 degrees and the performance of the HOG16 descriptors almost reaches chances level. The much worse performance of HOG16 may be related to its lower dimensionality (28) in comparison to the SIFT8 (128) and DAISY8 (200) descriptors.

We evaluated the performance of the descriptors on our virtual city dataset for keypoints in images taken under different viewpoint (Fig. 2) but under the same illumination using 1.3 million keypoint pairs. All images were taken at 1pm. The ranking for the descriptors was similar to that under changes in illumination (section 5.3) except for GLOH (Fig. 7). Under viewpoint changes, the performance of the GLOH8 descriptor is similar to that of SIFT8, not to DAISY8 as in section 5.3. This could be explained by the larger number of pooling regions in DAISY, 25 versus 17 in GLOH and 16 in SIFT. It appears that the arrangement of the pooling regions is important for illumination changes in the scene while the number of pooling regions matters in scenes captured from different viewpoints. Here, again the performance of HOG and SSIM descriptors may be related to the descriptor dimensionality.

5.5. Viewpoint and Illumination Change

In sections 5.3 and 5.4, we considered the effects of illumination change on a scene taken with a static camera and the effects of viewpoint change under constant illumination. Here, we compare the effects of camera position under different illumination for one of the descriptors DAISY8. The relative performance of the other descriptors was similar. We considered the performance of DAISY8 for scenes taken under different illumination (2, 4, 6, and 8 hours apart) with a static camera, with a camera at the same location at rotation stops of 22.5 degrees (Fig. 2 a)) and camera from different locations (Fig. 2 b)). The performance with the panning camera (Cam2) is similar to that of the static camera (Fig. 8). The task of matching keypoints in images taken from cameras at different location and orientation (Cam1) is a lot more challenging and the descriptor performance is considerably worse. This is because here the changes in perspective, occlusions, etc. play much larger role. It is especially true for keypoints around contour boundaries, where the background could significantly change due to changes in viewpoint.

5.6. 3D Descriptors

Depth can be acquired by many different means, at a range of quality levels. Since we know the full geometry of each scene in our virtual city, we have depth maps easily available (Fig. 4 c)), and we can assess the utility of incorporating depth information into feature descriptors. Since

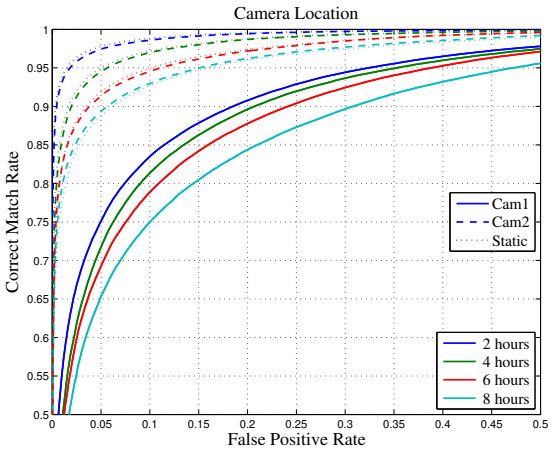

Figure 8. Performance of the DAISY8 descriptor for images of scenes under different illumination (2, 4, 6, and 8 hours apart) with a static camera, with a camera (Cam2) at the same location at rotation stops of 22.5 degrees (Fig. 2 a)) and a camera (Cam1) from different locations (Fig. 2 a)). The descriptor has most difficulty with large changes in viewpoint.

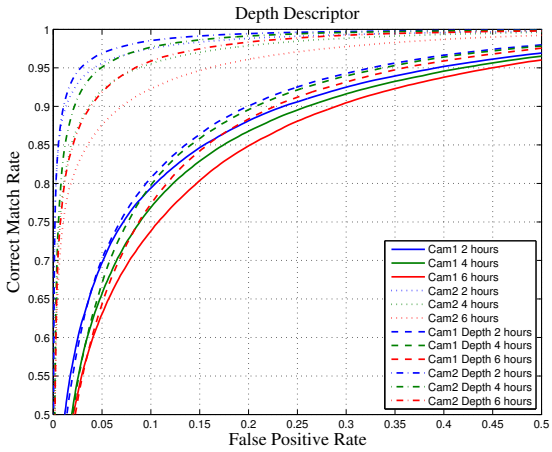

Figure 9. The performance of the SIFT8 descriptor in comparison with the combined SIFT8 on the RGB image plus the HOG16 on the depth map (64 depth levels) 3D descriptor under different camera viewpoint and varying illumination conditions. Note the performance of the 3D descriptor has a larger performance gain for larger changes in viewpoint (Cam1).

acquiring high resolution depth maps is difficult, we quantized the depth maps from our virtual city to n depth levels to approximate a depth map acquired in a real world setting. We expect that knowing depth will be particularly helpful in two scenarios. For images of a scene under different illumination, it can distinguish between edges due to depth discontinuities and due to shadows. For images under different viewpoint, it can help match keypoints on contour boundaries despite significant changes in the appearance of the background.

We propose to augment the feature descriptors in the following way. For each keypoint, we compute the descriptor, F_{rgb}, using the RGB image (Fig. 4 a)) and the descriptor, F_{depth}, using the depth map (Fig. 4 c)). Thus, the final descriptor is $[F_{rgb}; F_{depth}]$. We experimented with different combinations of descriptors for F_{rgb} and F_{depth} and different depth resolutions, $n = 16, 32, 64, 128$, and 256. We found that using descriptors based on histograms of oriented gradients for F_{depth} produced best results as they capture the information about the relative depth of the pixels in the neighborhood around the keypoint. To evaluate whether two keypoints match, we compute the weighted sum of the Euclidean distance between the descriptors from the RGB image, D_{rgb} and the Euclidean distance between the descriptors from the depth map, D_{depth}.

$$D_{desc} = \alpha D_{depth} + (1 - \alpha) D_{rgb}$$

We performed different experiments with various values of alpha. We see greater improvement in performance for larger changes in viewpoint and illumination. Figure 9 shows the performance of the SIFT8 descriptor for the RGB image, HOG16 descriptor for the depth map quantized to 64 depth levels and alpha value of 0.3 in comparison to using the SIFT8 descriptor alone. Even a very low dimensional descriptor as HOG16 (28) that adds minimal computational overhead produces a significant improvement in the performance of descriptors in challenging illumination and viewpoint conditions. Using higher dimensional descriptors like GLOH or SIFT for the depth map descriptor improves the performance further but at the expense of higher computational cost. Even depth maps with a resolution as low as 16 depth levels produce improvement in performance. Higher resolution depth maps (greater than 64 levels) improve the performance further but not significantly.

6. Conclusion

We used a photorealistic virtual world to evaluate the performance of image features. We used two datasets of photorealistic images –one from a virtual city and the other of a model of the Statue of Liberty. We showed that the performance of the descriptors on similar datasets from the real world and virtual Statue of Liberty is similar and results in the same ranking of the descriptors. Working in a virtual world allows complete knowledge of the geometry of the scene and full control of the environment, thus allowing to study the impact of different parts of the environment on the descriptors in isolation.

Our experiments on the dataset of our virtual city show that the DAISY descriptor performs best overall both under viewpoint and illumination changes. We found that spatial arrangement of the pooling regions in the gradient descriptors has an impact on the descriptor performance for matching keypoints in images taken under different illumination.

The number of pooling regions on the other hand needs to be considered for images taken from different camera viewpoint. The lower dimensional feature descriptors generally performed worse due to lack of distinctiveness. However, we showed that using a low dimensional descriptor such as HOG can help improve descriptor performance if applied to the depth map of the scene and used in conjunction with a feature descriptor over the RGB image. We ranked features with regard to specific image transformations (viewpoint, and lighting variations over time-of-day).

Using high quality 3D computer graphics models as we have here allows for controlled and specific evaluation of image features, and may allow new features to be designed and optimized for specific computer vision tasks.

7. Acknowledgments

This research was partially funded by Shell Research, Quanta Computer, ONR-MURI Grant N00014-06-1-0734, CAREER Award No. 0747120, ONR MURI N000141010933 and by gifts from Microsoft, Adobe, and Google.

References

[1] M. Brown and D. G. Lowe. Recognising panoramas. In *Proc. IEEE Int. Conf. Computer Vision*, 2003.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[3] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 264–271, 2003.

[5] V. Ferrari, T. Tuytelaars, and L. V. Gool. Simultaneous object recognition and segmentation by image exploration. In *Proc. European Conf. Computer Vision*, 2004.

[6] W. T. Freeman, D. Anderson, P. Beardsley, C. Dodge, H. Kage, K. Kyuma, Y. Miyake, M. Roth, K. Tanaka, C. Weissman, and W. Yerazunis. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, 18:42–53, 1998.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proc. IEEE Int. Conf. Computer Vision*, pages 649–655, 2003.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[10] D. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE Int. Conf. Computer Vision*, Sept. 1999.

[11] J. Marin, D. Vazquez, D. Geronimo, and A. M. Lopez. Learning appearance in virtual scenarios for pedestrian detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[13] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. In *Int. Journal of Computer Vision*, pages 800–807, 2005.

[14] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. Journal of Computer Vision*, 42(3):145–175, 2001.

[15] D. Parikh and C. L. Zitnick. The role of features, algorithms and data in visual recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010.

[16] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. IEEE Int. Conf. Computer Vision*, pages 754–760, Jan. 1998.

[17] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27:157–173, February 2008.

[18] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. Journal of Computer Vision*, 37:151–172, 2000.

[19] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[20] G. R. Taylor, A. J. Chosak, and P. C. Brewer. OVVV: Using virtual worlds to design and evaluate surveillance systems. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[21] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. 2008.

[22] TurboSquid. Library of 3D products. In *http://www.turbosquid.com/*, 2010.

[23] T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. British Machine Vision Conference*, pages 412–425, 2000.

[24] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[25] S. Winder and M. Brown. Learning local image descriptors. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[26] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[27] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D model matching with viewpoint-invariant patches (VIP). In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[28] J. Xiao, J. Hays, K. A. Ehinger, A. Torralba, and A. Oliva. SUN database: Large scale scene recognition from Abbey to Zoo. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010.