Hongwen Kang, Alexei A. Efros, Martial Hebert, Takeo Kanade, "Image Composition for Object Pop-out,"
The 2nd International IEEE Workshop on 3D Representation for Recognition (3dRR-09), September, 2009.

Image Composition for Object Pop-out

Hongwen Kang   Alexei A. Efros   Martial Hebert   Takeo Kanade
School of Computer Science
Carnegie Mellon University
{hongwenk, efros, hebert, tk}@cs.cmu.edu

Abstract

We propose a new data-driven framework for novel object detection and segmentation, or "object pop-out". Traditionally, this task is approached via background subtraction, which requires continuous observation from a stationary camera. Instead, we consider this an image matching problem. We detect novel objects in the scene using an unordered, sparse database of previously captured images of the same general environment. The problem is formulated in a new image composition framework: 1) given an input image, we find a small set of similar matching images; 2) each of the matches is aligned with the input by proposing a set of homography transformations; 3) regions from different transformed matches are stitched together into a single composite image that best matches the input; 4) the difference between the input and the composite is used to "pop-out" new or changed objects.

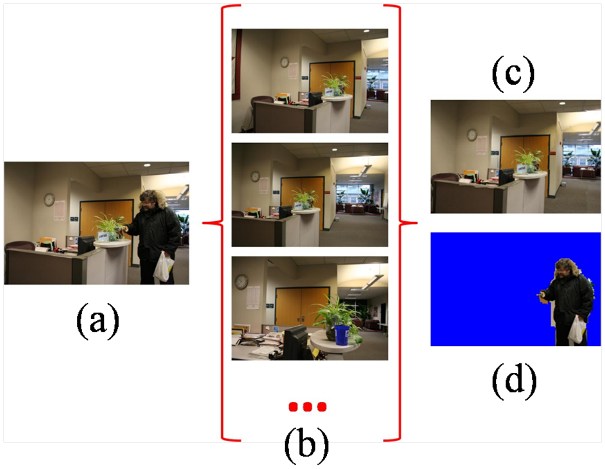


Figure 1. Given a single input image (a), we are able to "explain" it with bits and pieces of similar images taken previously (b), so as to generate a faithful representation of the input image (c) and detect the novel object (d).

1. Introduction

We are interested in tasks in which, given a single input image, we seek to "explain" it with bits and pieces of similar images taken previously, while detecting an interesting novel object that was not seen in prior images (Fig. 1). Locating objects within an image that might be of interest to a human is a very hard, severely under-specified task, yet it is something that we want our computers to be able to do. If we know what the person is looking for, then that task becomes quite a bit easier – many reasonably successful detectors exist for several classes of objects, such as cars and faces. However, what if the person is just looking for something "unusual" or "interesting". While low-level saliency methods can sometimes predict where a person might want to look, this rarely correlates with actual objects.

Movement has often been used as a way to detect interesting objects. Not only does movement gives out the boundaries of an object, but the very fact that it can change position indicates that it is worth paying attention to. In the context of a stationary camera, many approaches exist for detecting moving objects based on background subtraction variants (e.g., [13, 18, 22, 27]). This type of approaches has been most popular in surveillance scenarios where the objective is to detect new or anomalous patterns that have not been observed in the environment. However, this line of work has been heavily restricted by the dependence on the availability of stationary cameras covering the entire surveillance area, and continuously observing the foreground/background in order to generate a spatial-temporal background model.

The scenario that we are interested in this paper is to detect interesting objects within a known environment, but without relying on continuous observation of the environment by a dense set of cameras. Instead, we assume that we have a large amount of unordered images taken of the environment over a long period of time. One particular scenario is a mobile robot platform that can move around in an environment and capture images of its surroundings sparsely in spatial-temporal space. The goal of this work is to enable the detection and segmentation of interesting objects ("object pop-out") in new images of the environment. In this context, an interesting object is defined as one that has not been seen in the previously recorded images.

The core operation in our approach is image matching: given a new image, we search through the database to find similar images of the same location, and use this information to detect what has changed. This approach is inspired by the host of non-parametric scene matching methods that have recently become popular for mining large-scale datasets [8, 11, 24]. However, the fundamental difference is that these methods are only able to match a single whole image, so their performance can be only as good as their best match. And, in fact, their performance suffers tremendously for cases when good matches cannot be found. This is an important issue for our task since, on the one hand we need very good correspondences, but on the other hand, we are not likely to find images taken at exactly the same viewpoint, no matter how large our dataset is.

Instead, we propose to explain the input image as a *composite* of different pieces of images from the dataset, after applying the appropriate transformations. This will allow us not only to have a very faithful representation of the input image (Fig. 1c), but also let us be able to pop-out objects that are new or that have moved within the environment (Fig. 1d).

Given a database of pre-captured images of a particular large-scale environment, our approach is composed of the following steps: 1) image indexing and matching; 2) image alignment for composition element generation; and 3) image composition and outlier detection (see Fig. 2).

First the image database is indexed off-line for efficient retrieval. Given a new input image, a small set of similar reference images are retrieved. From these images, we perform image alignment to find all the reasonable transformations that could potentially warp the reference image to replicate the input image. For each reference image, multiple transformation proposals are generated. All of these proposed transformations are recorded as image composition elements.

The image composition step uses these proposals to explain the input image as a combination of different composition elements. The regions of the image that cannot be explained by *any* of the elements are declared outliers. These "popped-out" objects are the objects of interest.

The Photomontage and related work [1, 3] are most related to our work, in that they also use a stack of similar images to create a new composite image. However, the main goal of [1] is to create a composition that is visually better than any of the original ones and it has a lot of manual intervention (it is mainly an artist's tool). [3] uses only one reference image and assumes that the location of input paired images is known. One of our baseline algorithms is an adapted version of [3].

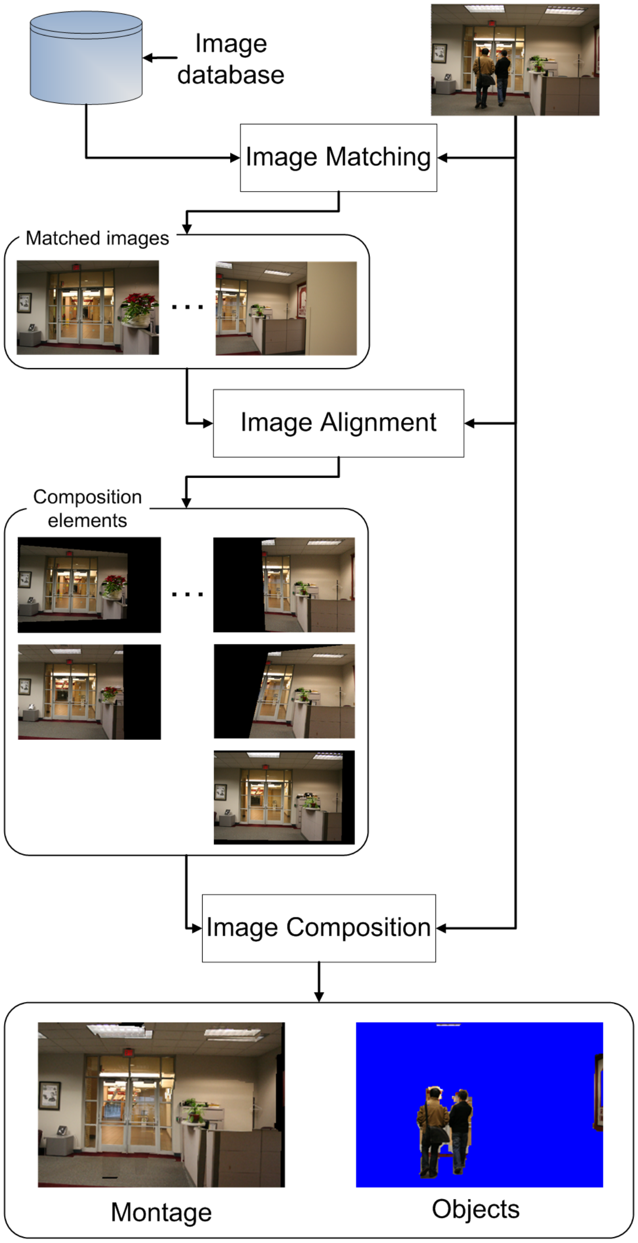


Figure 2. Processing diagram.

Multiple composition sources have also been used in [5] to detect novelty in the images. However, the problem they are solving is very different. They define irregularities as image patches that are different from other patches in the database (e.g., an apple in a field of oranges), while do not have any notion of scene geometry. Therefore, on our database, their algorithm will flag all objects seen from a different viewpoint as new, but miss all objects that have been simply moved.

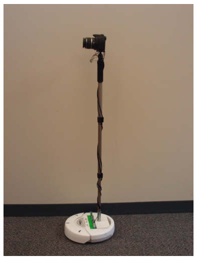


Figure 3. Data collection rig.

2. Our Approach

2.1. Dataset Acquisition

We build our database by using a fully-automated data collection rig. It uses an off-the-shelf iRobot roomba robot kit with a digital camera (Fig. 3). The robot explores the environment (the entire floor of an office building) autonomously, capturing still photographs approximately every 20 seconds. Data was captured over a year, with the goal of covering as much area and variance of the environment as possible. As a result, the whole database is composed of around $9,000$ images, and it contains information about what is usual for that environment and what is not.

2.2. Image Matching

Given an input image, we want to efficiently find a set of images that capture the same scene at the same location. Because within a typical indoor environment, things tend to be very self-similar and can be easily confused with one another, features for coarse scene recognition like the Gist [17] are not appropriate for this problem. Instead, we prefer features that are distinctive and robust to scale and viewpoint change. For this purpose, the combination of Hessian Affine (HESAFF) [14] region detectors and the SIFT [12] has shown superior performance with respect to distinctiveness and repeatability across view changes [15]. In searching for similar images from large databases, the analogy of images to text documents has introduced bag-of-words model and the use of text retrieval approaches in efficient image retrieval [9, 20]. Recently multi-level vocabulary trees [16] have been used for image retrieval to give an efficient way of searching for exact matched objects. Here, we learn a vocabulary tree with a fan-out factor of 4 and depth of 9 from a random sample of features in the database.

2.3. Image Alignment

For each matched image, we generate a number of images that are warped by homography transformation so as to align with the input image. Each of the warped image is called an image composition element (e.g., Fig. 2), used as input to the later image composition step. For that purpose, we use an iterative RANSAC based alignment algorithm [4, 25]. For each input/reference image pair, our image alignment algorithm takes N initially matched feature points as input. It fits a homography transformation model using the RANSAC algorithm, which finds N_i inliers and N_o outliers, with $N = N_i + N_o$. We initialize with all candidate matched features and recursively call the image alignment process using the matching outliers determined in previous rounds, until we have less than a certain number of matched feature points, or a maximum number of iterations is reached. In our experience, we found that using at least 30 initial matching points and at most 10 iterations gives the best balance of matching quality and computational efficiency. Also, we require a successful alignment to have at least 20 inliers. Using the aligned reference images, we generate a pool of image composition elements. In summary, the iterative RANSAC approach has three major benefits.

First, it filters out mis-matches from the image matching step. Because the bag-of-words model in the image matching algorithm discards all geometric information, it is very possible that the returned images have features which are visually similar to the input image, but are taken at different locations. They will be detected and removed by enforcing the geometric consistency through RANSAC homography estimation.

Second, a typical indoor environment could be approximated by multiple planes, and thus a single homography cannot satisfy all of the constraints. The iterative alignment algorithm works well in this scenario, in the sense that it finds all possible homography transformations between the input image and each matched database image.

Third, it makes the algorithm more robust to scenes with cluttered objects, e.g., the plants in Fig. 1. In this case, a large number of features are clustered in a small area. When RANSAC is applied, the cluttered object will have the most inliers of features, and homography transformation will be severely biased, if only one transformation is allowed.

In the literature, most successful work in image alignment and stitching use a single homography transformation, they are only able to work for images with planar scenes [7, 19, 23]. Another set of techniques do motion estimation using optical flow. While this works reasonably well in videos, it is only capable of small displacements, and is not applicable for our data.

2.4. Multilayer Image Composition

First, we focus on generating the composition of the input image using composition elements from the image alignment step. We denote the input image I, the set of image composition elements $\mathcal{R}_e$, and the target composition image I_t. At each pixel location p, the target image pixel p_t is a copy of one of the reference image pixels p_r, where $r \in \{1, \cdots, |\mathcal{R}_e|\}$. This is described in the following way,

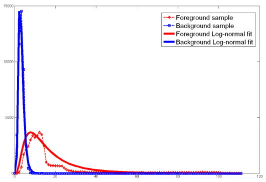


Figure 5. Comparing foreground (red) and background (blue) with statistics of minimum difference between input image and reference images.

$$
\begin{aligned}
p_t &= \sum_{r=1}^{\mathcal{R}_e} p_r \delta(l_p - r) \qquad (1)\\
where\ l_p, r &\in \{1, 2, \cdots, |\mathcal{R}_e|\}\\
\delta(l_p - r) &= \left\{ \begin{array}{ll} 1, & l_p = r \\ 0, & otherwise \end{array} \right. .
\end{aligned}
$$

The primary goal of image composition is to find a labeling (l_p) that generates a target composition which is visually as similar as possible to the input image. In addition, preserving neighborhood relationships is also important for visual consistency. We use a pairwise neighborhood constraint which penalizes any changes of labels between two neighboring pixels. Also, we apply another smoothness constraint which penalizes change of neighboring labels in uniform areas.

To solve this optimization problem, we build a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{A} \rangle$ to represent our target image I_t. $\mathcal{N}$ is the set of nodes, each of which corresponds to a pixel p_t, and $\mathcal{A}$ is the set of arcs which connect neighboring nodes. In our model we use a 4-neighborhood system.

Our labeling optimization problem is now formulated as minimizing the following energy function:

$$
E = \sum_{p \in \mathcal{N}} E_1(p) + \alpha \sum_{\substack{p \in \mathcal{N} \\ a_{pq} \in \mathcal{A}}} E_2(p, q) + \beta \sum_{\substack{p \in \mathcal{N} \\ a_{pq} \in \mathcal{A}}} E_3(p, q), \quad (2)
$$

where E_1 measures the unary energy (i.e. data term) coming from pixel-wise difference, and E_2, E_3 are both pair-wise term that preserves neighborhood structure. Finding the global minimum of this energy function is an NP hard problem. But the local optimal solution could be efficiently computed via the graphcut algorithm [1, 2, 3, 6, 10, 26].

2.5. Image Composition with Object Pop-out

We approach this object pop-out problem in a unified composition framework by introducing an extra "outlier" layer. Instead of assigning every target pixel a reference pixel, the algorithm has the option to decide that some pixels are beyond the representation power of the reference image and therefore could be potentially outliers, corresponding to the object that we want to pop-out. We model this outlier probability through an inlier-outlier model. The final decision of each target pixel's label is made through the graphcut optimization that combines both pixel-wise local information and neighborhood information. By denoting the outlier label as label 0, the range of l_p in (1) becomes $l_p \in \{0, 1, 2, \cdots, |R_e|\}$.

The algorithm is given a set of training samples, each of which consists of an input image with some known new objects and a set of reference images without the objects. If an object from the input image has not been changed, then we should be able to find its correspondence in at least one of the reference images. Therefore, we use the minimum distance between an input image and the reference images as our feature. Fig. 5 shows the probability density function of this feature for the inlier ("background") and the outlier ("foreground") classes, respectively. Specifically, we use the L_1 distance in CIELab color space.

For an input pixel p_i and corresponding pixel p_r in a reference layer, $r \in \{1, 2, \cdots, |R_e|\}$, the L_1 distance in CIELab space is $d(p_i, p_r)$, denoted by d_{ir}^p. We can fit this distance to our probabilistic inlier-outlier model. Instead of using image pixel color difference directly [3], we learn through training data the compatibility of p_i and p_r as the following cumulative likelihood:

$$
C_I(i, r, p) = P(d > d_{ir}^p | Inlier), \quad (3)
$$

and

$$
C_O(i, r, p) = P(d < d_{ir}^p | Outlier). \quad (4)
$$

An example of this conditional probability is shown in Fig. 5. The more similar the corresponding pixels are, the higher $C_I(i, r, p)$ and lower $C_O(i, r, p)$ will be.

Denoting the minimum distance between a pixel in the input image and the corresponding pixels in all the reference images as:

$$
d_m(i, p) = \min_{r \in \{1, 2, \cdots, R_e\}} d_{ir}^p \quad (5)
$$

we can calculate the likelihood that this pixel is an inlier as

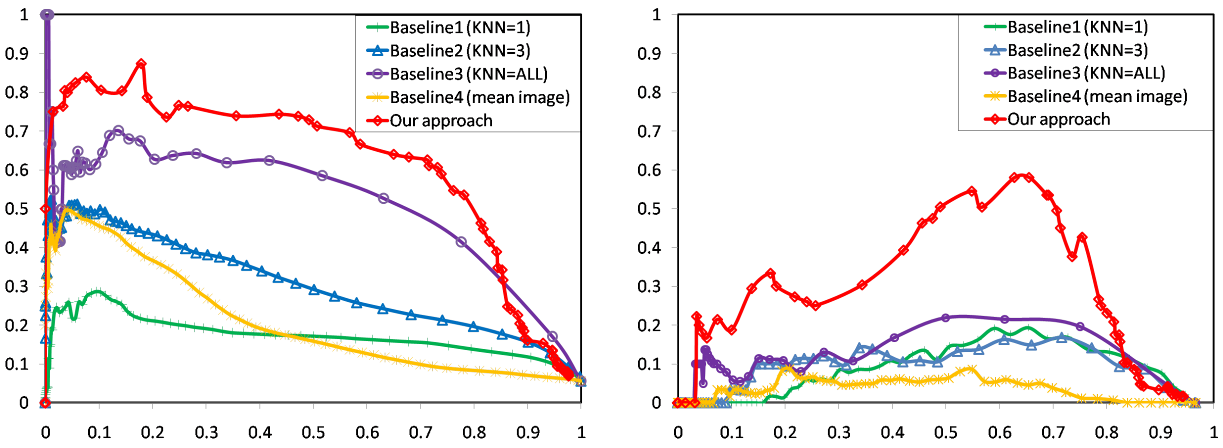
$$
C_I(i, p) = P(d > d_m(i, p) | Inlier), \quad (6)
$$

and also an outlier as

$$
C_O(i, p) = P(d < d_m(i, p) | Outlier). \quad (7)
$$

The unary term energy for labeling a target pixel as one of the layers $r \in \{0, 1, 2, \cdots, |R_e|\}$ is defined as

$$
E_1(p) = \sum_{r=1}^{R_e} \frac{C_O(i, r, p)}{C_I(i, r, p)} \delta(l_p - r) + W(i, p) \delta(l_p). \quad (8)
$$



(a)

(b)

Figure 4. (a) Pixel-wise precision-recall curve comparing our method with three baseline algorithms; (b) Detection hit/miss precision-recall curve comparing our method with three baseline algorithms.

In our experiment, we define $W(i, p)$ as a data-dependent parameter:

$$W(i, p) = \frac{C_I(i, p)}{C_O(i, p)}. \qquad (9)$$

$W(i, p)$ could also be a user-controlled parameter. Higher values of $W(i, p)$ lead to higher precision and lower recall. In fact, we use $W(i, p)$ in this way as a control variable in our numerical evaluation in order to generate the precision-recall characteristic curve. Although empirically we find that the data-dependent form as in (9) performs slightly better and does not require user tuning.

The first pairwise smoothness energy term is defined as:

$$\underset{a_{pq} \in \mathcal{A}}{E_2}(p, q) = 1 - \delta(l_p - l_q), \qquad (10)$$

which penalizes any change of labels at neighboring pixels.

We further emphasize the smoothness constraint based on the gradient of the input image by introducing the second smoothness energy term as:

$$\underset{a_{pq} \in \mathcal{A}}{E_3}(p, q) = (1 - \delta(l_p - l_q))e^{-|\nabla_{pq}(p)|}, \qquad (11)$$

where $\nabla_{pq}(p)$ is the input image gradient along the direction of the arch a_{pq}. Since we are using a 4 neighborhood system, this only penalizes changes of labels in the image row and column directions, especially at uniform regions.

After the objects are popped out, one may want to fill in the holes and generate the composition of the complete scene. There are various ways to do that under our framework. In this paper, we choose to fill in the missing portion using the reference image that covers the hole and requires the least scale, rotation and translation transformations based on our homography estimation.

3. Experimental Results

In this section we compare our proposed approach to four baseline algorithms. Our database is composed of about $9,000$ 1728×1152 images. For computational efficiency we down-scale the images to 640×427 during the image matching step. We noticed that the image matching quality decreases dramatically if the resolution is reduced further. After the composition elements are generated, we use a lower resolution of 320×214 in the graphcut optimization, since normally there are a large number of reference images to choose from, e.g., 50 to 100. The testing dataset is composed of 56 images with objects that are new or that have changed compared to the database images. Results are reported through 10-fold cross validation, and at each round we randomly select 30 images as training samples and we use the remaining images for testing. Graphcut parameters (α, β) were tuned once using 10 sample images from the testing dataset and then kept fixed through the testing. At each round, we estimate from the training images the distribution of the minimum difference between foreground and background pixels as shown in 5.

We use four baseline algorithms for comparison. For a fair comparison, we use reference images that have been aligned to the input image by using our image alignment algorithm. We measure the similarity of the input image and a reference image by how many feature points are matched between them, after the image alignment step. The baseline algorithms are:

- *1 nearest neighbor background subtraction* compares the input image with the best matched reference image. For each pixel p_i in the input image, its corresponding pixel p_r is looked up from the reference image and the difference between them is calculated in CIELab color space. Then a threshold is used to determine which

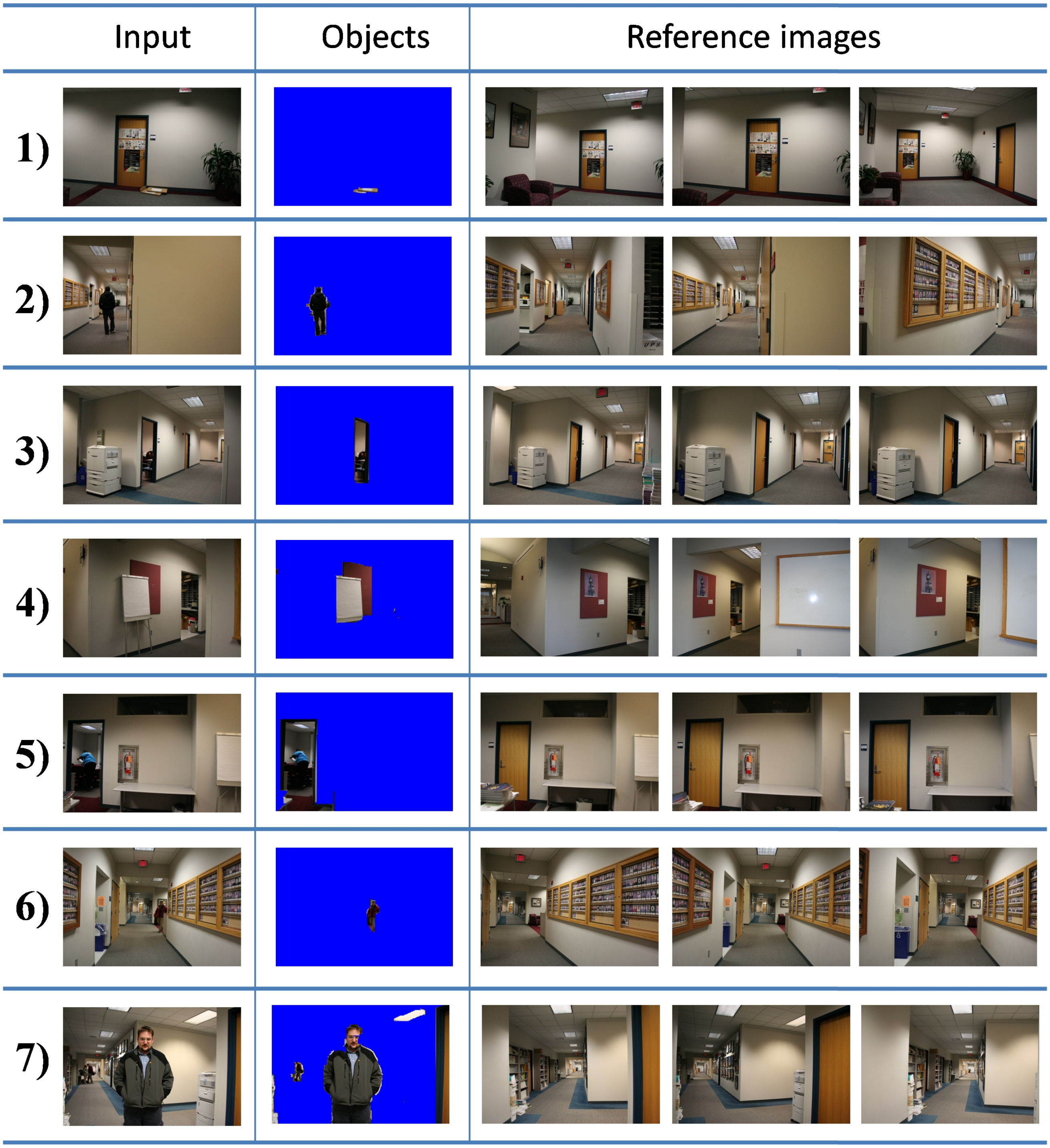


Figure 6. Some qualitative examples of our approach working under various locations of the large office environment, popping out objects that are new or changed at various size, shape and illumination condition.

part of the scene has changed. By changing this threshold, we can vary the characteristic of our detector. This is in fact a simplified version of [3]. The major difference is that in [3], 1) the goal was registering pairs of images; 2) the location of input paired images is known, while we propose a framework that uses image matching to search through a large-scale database (Sec. 2.2); 3) instead of showing only a few qualitative images as in [3], we carried out large-scale quantitative experiments; 4) [3] formulates the disparity in fundamental matrix, and we found that when more reference images are available the performance improves significantly, and a simpler homography model is sufficient.

- *K nearest neighbors (KNN) background subtraction* is similar to the first baseline algorithm, with K reference images instead of one. The minimum distance between each pixel in the input image and corresponding pixels in reference images is calculated, and the ones greater than a threshold are detected as changes. Choosing $K = 3$ gives us the second baseline for comparison.
- *$K = ALL$ nearest neighbors*. Extending K to the number of all matched images gives us the third baseline algorithm. The difference between this baseline and our approach is that it does not use foreground/background models and spatial constraints.
- *Mean image subtraction*. The mean of many matched images could be viewed as a background model for the environment captured by the input image. The difference between the input image and the mean image can then be used to detect changes in the scene. Here, we generate the mean image from the entire set of aligned reference images, and the pixels with difference higher than a threshold are output as changes.

The control variable for our approach is a uniform prior weight W in (8), higher values of W favor higher precision and lower recall. The first performance measure we use is a pixel-wise precision and recall. It measures how many pixels are correctly classified. We observe from Fig. 4(a) that using more reference images and adding background/foreground, spatial constraint help in boosting the performance.

In the second performance measure, a result is counted as a correct detection if the area of the intersection of the detected region and the ground-truth object region is at least 50% of each of them. By measuring how well the detected region matches the object, this measure is stricter in that it penalizes heavily algorithms that detect groups of small patches scattered across the whole image, which is the case for all the baseline algorithms that do not use neighborhood information. Also note that the precision goes to zero for all three baseline methods when using the second measure, it is because that when we continue increasing the detection threshold, none of the output regions are correct (zero precision, zero recall), i.e. if a weak detector (in our case, the baseline methods) predicts N detections and all are false alarms, it has zero precision, zero recall.

In Fig. 6 and 7 we show more qualitative examples of our approach at different locations of the office environment, popping out objects that are either new or changed with various sizes, shapes and illumination conditions. Fig. 8 shows a typical failure case of our algorithm when the input image has illumination that is dramatically different from the database images.

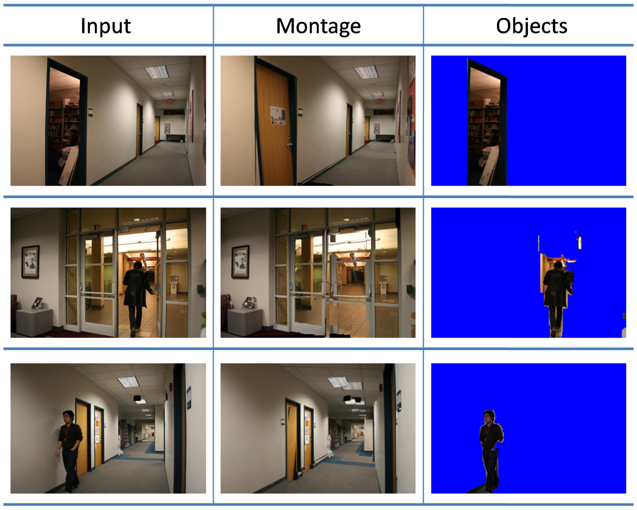
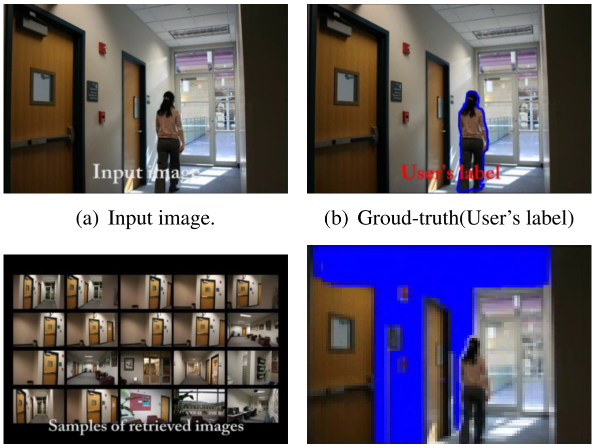


Figure 7. Example results of montage and objects pop-out.



(a) Input image.

(b) Groud-truth(User's label)

(c) Retrieved images.

(d) Our result

Figure 8. Our approach fails when the input image has illumination that is dramatically different from the database images.

4. Conclusions

We proposed a data-driven framework for novel object detection and segmentation, or "object pop-out". We detect novel objects in the scene by using an unordered, sparse database of previously captured images of the same general environment. We demonstrated the effectiveness of our approach in detecting changed objects, as well as providing faithful representation of the background. There are several limitations in our current approach that we would like to strengthen further. First, our color based feature can be strengthened further on its robustness to large illumination changes. Second, in our current system, temporal information is not explicitly used, which could be useful in determining the order of the environment changes. Also, it will also be of interest to apply our approach to outdoor scenes, such as community image datasets [21].

Acknowledgements

The authors would like to thank anonymous reviewers for helpful and constructive suggestions. This material is based upon work partially supported by the National Science Foundation under Grant No. EEC-0540865.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, 2004.

[2] S. Bagon. Matlab wrapper for robust higher order potentials, January 2009.

[3] P. Bhat, K. C. Zheng, N. Snavely, A. Agarwala, M. Agrawala, M. F. Cohen, and B. Curless. Piecewise image registration in the presence of multiple large motions. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2491–2497, Washington, DC, USA, 2006. IEEE Computer Society.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63(1):75–104, 1996.

[5] O. Boiman and M. Irani. Detecting irregularities in images and in video. *Int. J. Comput. Vision*, 74(1):17–31, 2007.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001.

[7] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, pages 1218–1225, Nice, 2003.

[8] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[9] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In A. Z. David Forsyth, Philip Torr, editor, *European Conference on Computer Vision*, LNCS. Springer, oct 2008. to appear.

[10] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[11] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT Flow: dense correspondence across different scenes. In *Proceedings of the 10th European Conference on Computer Vision, Marseille, France*, 2008.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[13] J. Migdal, T. Izo, and C. Stauffer. Moving object segmentation using super-resolution background models. In *Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, 2005.

[14] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

[16] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

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

[18] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *IEEE International Conference on Computer Vision*, Kyoto, Japan, 2009. IEEE Computer Society.

[19] H.-Y. Shum and R. Szeliski. Correction to construction of panoramic image mosaics with global and local alignment. *Int. J. Comput. Vision*, 48(2):151–152, 2002.

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[21] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press.

[22] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:259, 2003.

[23] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.

[24] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[25] J. Y. A. Wang, Edward, and H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3:625–638, 1994.

[26] O. J. Woodford, I. D. Reid, and A. W. Fitzgibbon. Efficient new view synthesis using pairwise dictionary priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis*, 2007.

[27] Y. Zhou, W. Xu, H. Tao, and Y. Gong. Background segmentation using spatial-temporal multi-resolution mrf. In *WACV-MOTION '05: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05) - Volume 2*, pages 8–13, Washington, DC, USA, 2005. IEEE Computer Society.