Change Detection from a Street Image Pair using CNN Features and Superpixel Segmentation

Ken Sakurada
sakurada@vision.is.tohoku.ac.jp
Tohoku Univeristy
6-6-01 Aramaki Aza Aoba, Aoba-ku, Sendai-shi, Miyagi, Japan

Takayuki Okatani
okatani@vision.is.tohoku.ac.jp
Tohoku Univeristy
JST, CREST
6-6-01 Aramaki Aza Aoba, Aoba-ku, Sendai-shi, Miyagi, Japan

Abstract

This paper proposes a method for detecting changes of a scene using a pair of its vehicular, omnidirectional images. Previous approaches to the problem require the use of a 3D scene model and/or pixel-level registration between different time images. They are also computationally costly for estimating city-scale changes. We propose a novel change detection method that uses features of convolutional neural network (CNN) in combination with superpixel segmentation. Comparison of CNN features gives a low-resolution map of scene changes that is robust to illumination changes and viewpoint differences. Superpixel segmentation of the scene images is integrated with this low-resolution map to estimate precise segmentation boundaries of the changes. Our motivation is to develop a method for detecting city-scale changes, which can be used for visualization of damages of a natural disaster and subsequent recovery processes as well as for the purpose of maintaining/updating the 3D model of a city. We have created a dataset named *Panoramic Change Detection Dataset*, which will be made publicly available for evaluating the performances of change detection methods in these scenarios. The experimental results using the dataset show the effectiveness of our approach.

1 Introduction

This paper proposes a method for detecting changes of a scene using a pair of its vehicular, omnidirectional images. Figure 1 shows an example of such image pairs taken at different times. Apparently, there are temporal differences in illumination and photographing conditions. Moreover, there has to exist visual difference in camera viewpoints, although they were captured from a vehicle running on the same street and were matched using GPS data. This is due to differences in vehicle paths and shutter timing. The type of scene changes targeted here includes 3D (e.g. vanishing/emergence of buildings, cars etc.) as well as 2D changes (e.g. changes of textures on building walls). To precisely detect these changes from such an image pair, it is necessary to overcome these unwanted visual differences.

Pages 61.1-61.12
DOI: https://dx.doi.org/10.5244/C.29.61

Figure 1: Example of an image pair of a scene captured two months apart.

To cope with these issues, some of the previous studies consider the problem in the 3D domain. They assume that a 3D model of a scene is given beforehand or can be created from images, and that the input images can be registered to the model with pixel-level accuracy [14, 20, 24]. However, a 3D model is not always available for every city. Besides, it is sometimes hard to perform precise image registration, due to lack of sufficient visual features. These are particularly the case when the scene undergoes enormous amount of changes. Working in the 3D domain tends to require large computational cost, which can be another difficulty when we want to detect changes for a large city.

Thus, we tackle the change detection problem in the 2D domain. That is, we consider detecting changes based on the direct comparison of a pair of images. The major issue is then how to deal with the above unwanted visual differences (i.e., viewpoint differences etc.) To cope with this, we propose to use the features extracted by convolutional neural networks (CNNs). To be specific, we use a fully trained CNN for large-scale object recognition task [11] in a transfer learning setting. It was reported in the literature that using activation of the upper layers of a CNN trained for a specific task can be reused for other visual classification tasks. Several recent researches imply that the upper layers of CNNs represent and encode highly-abstract information about the input image [2, 3, 27]. We conjecture that highly-abstract (or object-level) changes can be detected by using the upper layers, whereas low-level visual changes (e.g. edge, texture etc.) will be detected using the lower layers. We show that this conjecture is true through several experimental results.

As will be shown below, CNN features indeed can detect the occurrence of scene changes accurately in the presence of the above unwanted image changes. However, they cannot provide precise segmentation boundaries of the changes by their nature. Thus, our method integrates the CNN features with superpixel segmentation of the input images. To be specific, the method first divides the input images into coarse grids, and estimates the likelihood of scene changes at each grid cell by comparing the CNN features of the grid cell at different times. Next, the method projects the detected changes at the grid cells into superpixel segments to obtain precise boundaries of the changes. The outline is shown in Fig. 2.

The motivation behind this study is to develop a change detection method that can be used for visualization of damages of a natural disaster and subsequent recovery processes as well as for the purpose of maintaining/updating the 3D model of a city. The main application scenario of the method is as follows. A vehicle with an omnidirectional camera and a GPS sensor on its roof is driven on every street of a city twice some time apart, yielding two sets of a large number of omnidirectional street images. Then, two images from each of the two sets are paired that are captured from the closest viewpoints by using the GPS data. Each image pair is inputted to the proposed method to detect changes for the pair, resulting in the estimation of changes over the entire city. To evaluate the method, we have created a dataset named *Panoramic Change Detection Dataset*, which will be made publicly available for evaluating the performances of change detection methods in this scenario.

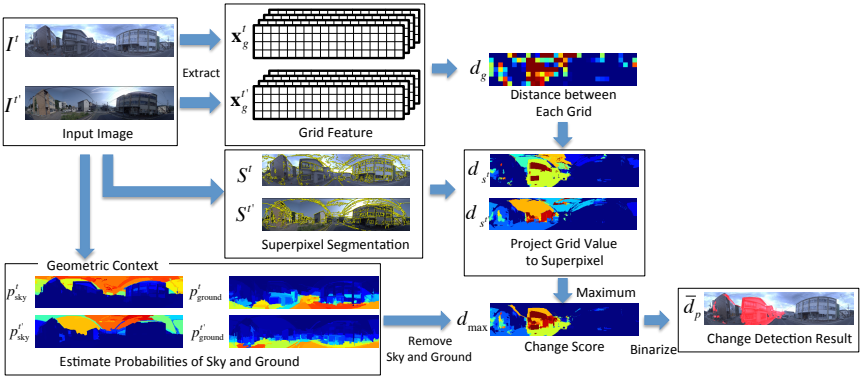


Figure 2: Flowchart of the proposed change detection method.

2 Related Work

There are a large amount of studies on the problem of detection of scene changes [1, 4, 5, 9, 10, 17, 18, 22, 28, 29]. They can be classified into several categories depending on type of scene changes to detect, methods, available information etc.

A standard approach is to detect changes in the 2D (image) domain [17, 19]. A typical method is to create an appearance model of a scene from a set of its images captured at different times, against which it compares a newly captured query image to detect changes. The major concern in this type of studies is with how to deal with irrelevant appearance changes such as difference in illumination. It usually requires the images to be captured from the same viewpoint, and thus cannot deal with query images captured from different viewpoints.

There are studies that formulate the problem in the 3D domain [5, 8, 9, 17, 23]. They build a model of the target scene in a "steady state," and compare a query image against it to detect changes. A 3D model of the scene is often created by using a 3D sensor other than cameras. In [8], to estimate the existence of a building, the edges extracted from its aerial images are matched with the projection of its 3D model to detect changes. The studies of Taneja et al. [23, 24] are classified into the same category. Their method is designed for maintenance/updating of an existing 3D model of a city. The scenario is that the method is used to detect scene changes in a low-cost manner, narrowing down the part to be remeasured by a 3D sensor to update the model.

There is another type of studies, where a large number of multi-view images of a scene are used to create its spatio-temporal model by leveraging the method of structure from motion. Schindler et al. proposed a method that uses a large number of images of a city that were taken over several decades [20]. Their method can perform several types of temporal inferences, such as estimating the time when each building was constructed. The recent work of Matzen and Snavely [14] is similar to that of Shindler et al. in the spirit. Their method uses internet photo collections to detect 2D changes of a scene, such as changes of advertisements and painting on a building wall. Assuming that a sufficient number of multi-view images of a scene are available, both methods use SfM to reconstruct 3D models of a scene [1, 9, 18, 22, 28, 29].

The method proposed in this paper falls in the class of 2D methods. It compares a pair of (omnidirectional) images to detect scene changes. As these images are captured at every

several meters from a vehicle running on a street, their viewpoints can be several meters apart in the worst case. Assuming that the image pair is aligned up to this accuracy (by using GPS data), the method manages to distinguish changes to be detected from irrelevant changes due to the difference in viewpoint, illumination and photographing condition. It does not need a dense 3D model of a scene, or a large number of images to perform SfM to build a 3D model.

3 Change detection using grid features

Figure 2 illustrates the outline of the proposed method. It consists of the three components: i) extraction of grid features, ii) superpixel segmentation, and iii) estimation of sky and ground areas by Geometric Context. These are described below.

(i) Extraction of grid features We denote two input images by I^t and $I^{t'}$, where t and t' are the times at which they were captured. First, I^t and $I^{t'}$ are divided into grid cells $g(=1,...,N_g)$. A feature is extracted from each grid cell g, yielding $\mathbf{x}_g{}^t$ and $\mathbf{x}_g{}^{t'}$.

The changes that we want to detect are object-level changes (e.g, the emergence/vanishing of buildings and cars) and not low-level, appearance changes due to changes in viewpoints, illumination or photographing conditions. To distinguish these two, the proposed method uses the activation of a upper layer of a deep CNN for the grid features $\mathbf{x}_g{}^t$ and $\mathbf{x}_g{}^{t'}$. To be specific, we use a pooling layer of the CNN. Each feature (e.g., $\mathbf{x}_g{}^t$) is the activation of all the units in the same location across the maps of the pooling layer. Thus $\mathbf{x}_g{}^t$ has the same number of elements as the maps of the pooling layer. The details are described in Section 4.

Next, these features are normalized so that $\|\mathbf{x}_g{}^t\|_2 = 1$, and then their dissimilarity is calculated at each grid cell g as

$$d_g = \|\mathbf{x}_g{}^t - \mathbf{x}_g{}^{t'}\|_2. \tag{1}$$

Then, the dissimilarity d_g is projected onto the input images I^t and $I^{t'}$, determining the pixel-level dissimilarity $d_p(p=1,...,N_p)$; N_p is the number of pixels. This is done by simply setting $d_p = d_g$ for any pixel p contained in the grid cell g.

(ii) Superpixel segmentation The difference in viewpoint is arguably the major source of difficulties for 2D change detection methods. The use of the CNN features is expected to help mitigate this difficulty, owing to the property of the CNN features invariant to geometric transformation such as translation, 3D rotation, and even more complicated ones. However, the resolution of the dissimilarity map d_p's is basically very low. We use superpixel segmentation to refine the dissimilarity map to hope for obtaining precise boundaries of the detected changes.

This starts with computing superpixel segmentation of I^t and $I^{t'}$. Let s^t be a superpixel and S^t be the set of superpixels. We define the dissimilarity d_{s^t} at a superpixel $s^t \in S^t$ to be the average of all the pixels in s^t as

$$d_{s^t} = \frac{1}{|s^t|} \sum_{p \in s^t} d_p. \tag{2}$$

We denote the maximum value of d_{s^t} and $d_{s^{t'}}$ by $d_{\max}$, i.e., $d_{\max} = \max(d_{s^t}, d_{s^{t'}})$.

(iii) Estimation of sky and ground areas by Geometric Context In the last step of the proposed method, Geometric Context [6] is used to remove the segments of sky and ground from the images. Geometric Context is a segmentation method that is known to be robust to changes in illumination and photographing conditions. It estimates probabilities of the sky and the ground at each pixel $({p_{\text{sky}}}^t, {p_{\text{ground}}}^t)$ in the input image I^t. Using these, we remove these areas from the images, converting the dissimilarity at each pixel into $\overline{d_p}$ as

$$\overline{d_p} = \begin{cases} 0 & (((p_{\text{sky}}{}^t > a) \wedge ({p_{\text{sky}}}^{t'} > a)) \\ & \vee (({p_{\text{ground}}}^t > b) \wedge ({p_{\text{ground}}}^{t'} > b))), \\ d_{\max} & (\text{otherwise}) \end{cases} \tag{3}$$

where $a = t_{\text{sky}}$ and $b = t_{\text{ground}}$ are constant values within the range of $0 \leq t_{\text{sky}}, t_{\text{ground}} \leq 1$.

4 CNN layer activation as grid features

When applied to object recognition, CNNs are robust (i.e., invariant) to differences in viewpoints and illumination condition, and nevertheless are sensitive to highly-abstract, semantic differences of images. This property is the key to the success of CNN for object category recognition. The reason why we employ CNN features is that we expect this property will also be useful for our problem. That is, we expect that if a scene has not changed, then its image feature should not change even when the viewpoints or illumination conditions are slightly different between the times of image acquisition; and the converse is true.

We choose one of the state-of-the-art CNN (known as the VGG net) for image recognition, proposed by Simonyan and Zisserman [21]. It has sixteen layers, from which we select one of its five pooling layers. It has convolutional layers with stride one, so that the spatial resolution does not change before and after each convolution layer. The feature of each grid is normalized so that its vector has length one, as mentioned above. As the CNN uses the rectified linear units (ReLUs) [11], the feature vector will have non-negative values. Hence, each element d_i of the dissimilarity $\mathbf{d}_g$ between different time images is within the range $[0, \sqrt{2}]$.

We consider two models of this CNN trained for two different tasks. One is a model trained for large scale object recognition of ILSVRC. The other is a model trained for a scene classification task using the SUN dataset [30]. To be specific, starting from the above fully-trained model for ILSVRC, we retrain it using the SUN dataset. The idea is that we may expect the improvement of performance by tuning the CNN to more relevant task for our purpose.

Instead of the CNN features, any feature may be used for the grid feature. The next section compares the CNN feature against Dense-SIFT [12] and local patch features (raw pixel brightnesses). Before CNNs were found to perform well for object recognition, Bag-of-Visual Words (BoVW) and Fisher vector, which encode the population of local features such as SIFT, are the former state-of-the-art [12, 13, 15, 16, 25, 26].

5 Experimental results

5.1 Panoramic change detection dataset

We have created *Panoramic Change Detection Dataset* and used it for the experiments[1]. The dataset consists of two subsets, named "TSUNAMI" and "GSV." "TSUNAMI" consists of one hundred panoramic image pairs of scenes in tsunami-damaged areas of Japan. "GSV consists of one hundred panoramic image pairs of Google Street View. The size of these images is 224×1024 pixels.

For each of them, we hand-labeled the ground truth of scene changes. It is given in the form of binary image of the same size as the input pair of images. The binary value at each pixel indicates that a change has occurred at the corresponding scene point on one of the paired image (which we call the query image). We defined the scene changes to be detected as 2D changes of surfaces of objects (e.g., changes of the advertising board) and 3D, structural changes (e.g., emergence/vanishing of buildings and cars). The changes due to differences in illumination and photographing condition and those of the sky and the ground are excluded, such as changes due to specular reflection on building windows and changes of cloud and signs on the road surface. The differences in viewpoint and illumination make it difficult to judge the existence of changes even by human vision. In fact, it took twenty minutes on average for an annotator to create the ground-truth map for an image pair. This demonstrates the necessity of a method for detecting scene changes automatically and accurately.

5.2 Detailed experimental configuration

In the proposed method, there are several parameters: (1) threshold t_{dist} for the grid feature used to binarize the detected changes, (2) thresholds used for Geometric Context to detect sky and ground (t_{sky}, t_{ground}), and (3) the parameters of superpixel segmentation.

In the experiments, the thresholds t_{dist} are determined by 5-fold cross-validation using the change detection dataset (Table 1). In the case of pooling-layer features and Dense-SIFT, $d_i \in \mathbf{d}_g$ takes a value within the range of $0 \leq d_i \leq \sqrt{2}$ because all elements of the features are non-negative values. In the case of gray-scale local-patch, d_i takes a value within the range of $0 \leq d_i \leq 2$. The thresholds of pool-3, 4, 5 and gray-scale local-patch are almost the median values of their range. As for the thresholds for Geometric Context, they are fixed for all the experiments as $t_{\text{sky}} = 0.2$ and $t_{\text{ground}} = 0.8$.

For the superpixel segmentation, we used Felsenszwalb's method of efficient graph based image segmentation [7]. The parameters of the superpixel segmentation (scale, diameter of a Gaussian kernel, minimum component size) are fixed for all experiments.

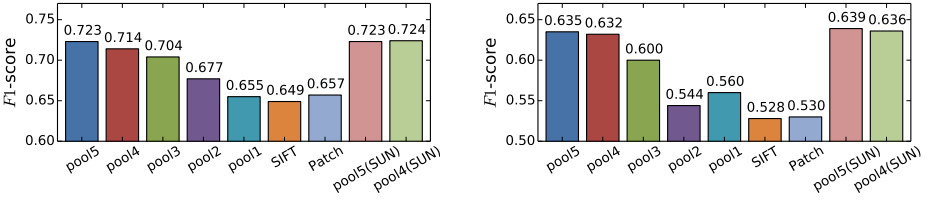
As mentioned earlier, we use two CNN models having the same structure as Simonyan-Zisserman [21] trained for object category recognition (ILSVRC) and scene classification (SUN).

For the sake of comparison, we used Dense-SIFT [12] and gray-scale local patch for grid features. These features are generated independently for each grid cell. We chose the same grid size as that when using the pool-5 layer. For Dense-SIFT, we extract descriptors at four different scales whose basic size is the grid size, and concatenate them to form a feature

[1]The data used in this study (the pairs of the omnidirectional panoramic images taken at different time points and the hand-labeled ground-truth of change detection) are available from our web site: http://www.vision.is.tohoku.ac.jp/us/download/

Table 1: Thresholds t_{dist} determined by 5-fold cross-validation using Panoramic change detection dataset "TSUNAMI" and "GSV".

	pool5	pool4	pool3	pool2	pool1	Dense SIFT	Patch	pool5 (SUN)	pool4 (SUN)
TSUNAMI	0.75	0.75	0.71	0.64	0.35	0.24	0.83	0.76	0.86
GSV	0.75	0.78	0.72	0.65	0.58	0.24	0.92	0.76	0.88



(a) TSUNAMI (b) GSV

Figure 3: F_1 scores of the change detection by various methods (average of 100 images in TSUNAMI and GSV dataset). 'pool-x' indicates the features extracted from the corresponding pooling layer of the CNN trained for the ILSVRC object recognition. 'pool-x(SUN)' indicates similar features from the CNN trained for a scene classification task using the SUN database.

vector. That is, a grid feature of Dense-SIFT has $128 \times 4 = 512$ elements. Local patch feature is the set of the raw, gray scale pixels within a grid cell. Each grid cell is resized to 16×16 pixels, generating a feature vector of of 256 elements.

5.3 Comparison of the results

Figure 3 shows the F_1 scores obtained when each feature is used for the grid feature. It is seen from the figure that the features of pool 4 and 5 achieve the best F_1 scores. The retraining of the CNN for a scene recognition task does not improve the results much. It is also seen that upper pooling-layers perform better than lower pooling-layers. F_1 scores of pool 1 is almost the same as that of the baseline methods (Dense-SIFT, gray-scale local-patch). These results validate the use of the CNN feature for our change detection problem.

Comparing the results for "TSUNAMI" and "GSV," the latter is worse in terms of accuracy. We think that this is because (i) some image pairs have too large a viewpoint difference to be dealt with by the grid size we used, and (ii) conversely, most of the scene changes are too small to be dealt with by the grid size. These two are contradict with each other, and it is not easy to resolve this contradiction immediately.

Figures 4 and 5 show examples of the result of change detection. (**See the supplementary note for other results.**) It is observed from them that the proposed method was able to correctly detect the scene changes, for example, demolished and new buildings, cars and debris. In some cases, Geometric Context could not accurately estimate sky due to electrical wire and pole, or could not distinguish between the ground and low height object (e.g., debris and car). The proposed method was nevertheless able to detect object-level scene changes fairly accurately.

We also compared the results obtained when using different pooling layers of the CNN for grid features, as shown in Fig.6. Note that the grid size for each result is determined by the number of units in a chosen pooling layer. The results show that the feature of upper

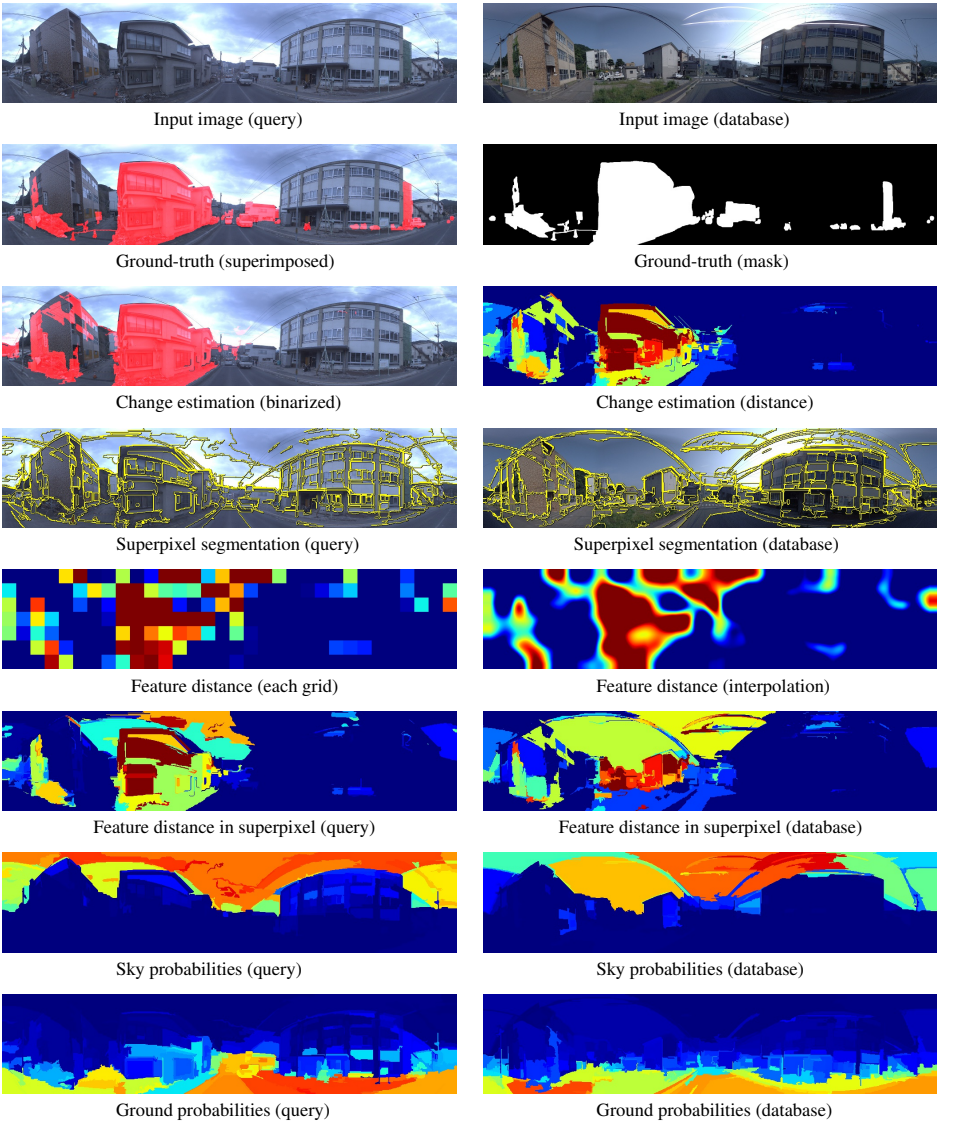


Figure 4: Results of change detection using pool-5 feature of CNN (Frame No. 1/100 in change detection dataset of tsunami)

pooling-layer discriminates highly abstract, object-level differences in the scene, whereas that of lower pooling-layer detects the difference of low-level visual feature (e.g., edges). This tendency confirms our conjecture described earlier. Furthermore, it implies that we could improve estimation accuracy by adaptively choose layers depending on the abstraction level of interest.

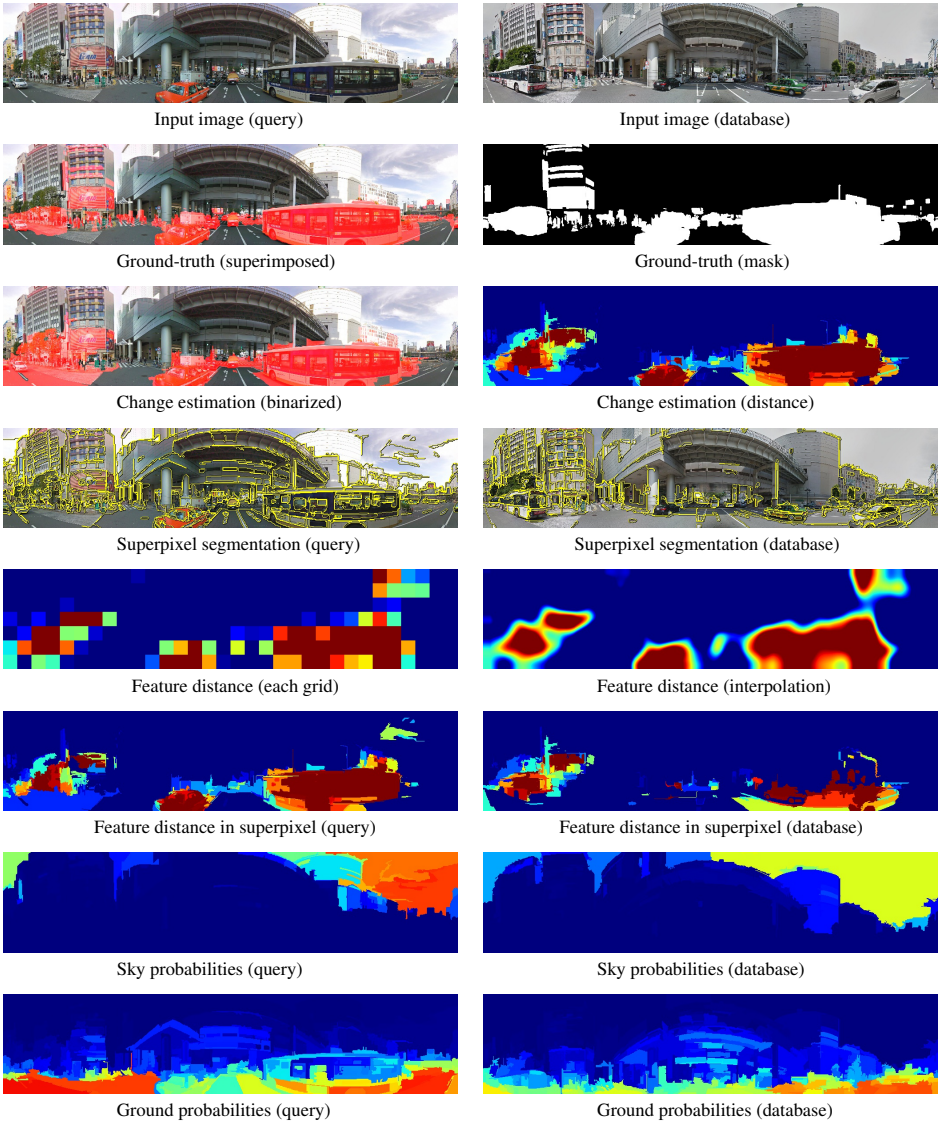


Figure 5: Results of change detection using pool-5 feature of CNN (Frame No. 1/100 in change detection dataset of google street view)

6 Summary

This paper has described a novel method for detecting temporal changes of a scene from a pair of its images. To cope with differences in viewpoint, illumination etc. between different times, the method uses activation of an upper layer of a convolutional neural network, which are expected to be invariant to the appearance changes caused by such differences. To recover spatial resolution lost by the pooling operations of the CNN, the method integrates the CNN features with superpixel segmentation of the scene images. For the purpose of experimental evaluation, we have created a dataset named *Panoramic Change Detection Dataset*

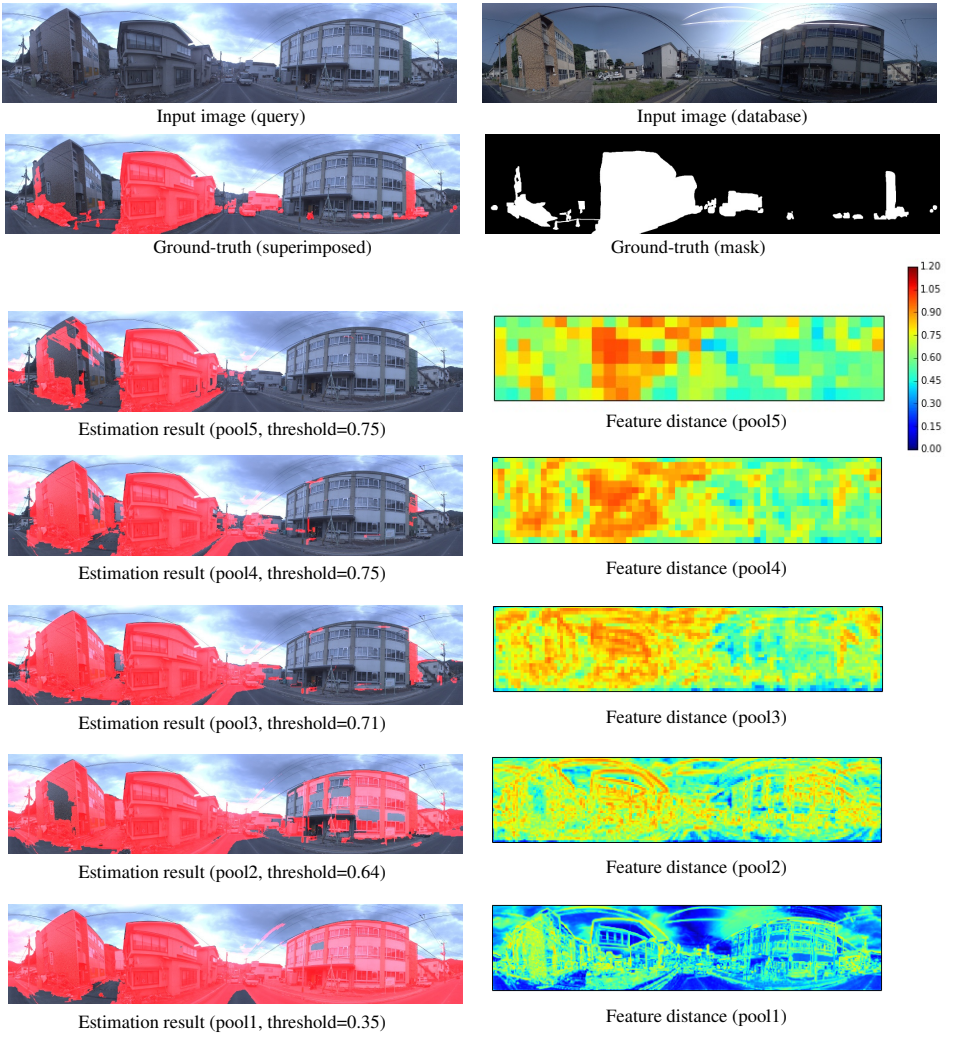


Figure 6: Feature distance of each grid (pooling-layers of CNN). Distance of normalized features between each grid $d_i \in \mathbf{d}_g$ takes a value within the range of $0 \leq d_i \leq \sqrt{2}$ because all elements of pooling-layer feature are non-negative values.

which includes images taken at tsunami-damaged areas and Google Street View images. The experimental results obtained using the dataset show the effectiveness of the proposed approach.

Acknowledgement

This research is supported by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan) and by MEXT KAKENHI Grant Number 25280054.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building Rome in a day. In *ICCV*, pages 72–79, 2009.

[2] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *ECCV*, pages 584–599. Springer, 2014.

[3] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] David Crandall, Andrew Owens, Noah Snavely, and Daniel Huttenlocher. Discrete-Continuous Optimization for Large-Scale Structure from Motion. In *CVPR*, pages 3001–3008, 2011.

[5] Daniel Crispell, Joseph Mundy, and Gabriel Taubin. A Variable-Resolution Probabilistic Three-Dimensional Model for Change Detection. *Geoscience and Remote Sensing*, 50(2):489–500, 2012.

[6] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, pages 654–661, 2005.

[7] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[8] Andres Huertas and Ramakant Nevatia. Detecting Changes in Aerial Views of Man-Made Structures. In *ICCV*, pages 73–80, 1998.

[9] David Cooper Ibrahim Eden. Using 3D Line Segments for Robust and Efficient Change Detection from Multiple Noisy Images. In *ECCV*, pages 172–185, 2008.

[10] Jana Košecka. Detecting changes in images of street scenes. In *ACCV*, pages 590–601. Springer, 2012.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[12] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, pages 28–42. Springer, 2008.

[13] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60 (2):91–110, 2004.

[14] Kevin Matzen and Noah Snavely. Scene chronology. In *Proc. European Conf. on Computer Vision*, 2014.

[15] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, pages 1–8. IEEE, 2007.

[16] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156. Springer, 2010.

[17] Thomas Pollard and Joseph L. Mundy. Change Detection in a 3-d World. In *CVPR*, pages 1–6, 2007.

[18] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed Real-Time Urban 3D Reconstruction from Video. *IJCV*, 78(2-3):143–167, 2008.

[19] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image Change Detection Algorithms: A Systematic Survey. *Transactions on Image Processing*, 14(3):294–307, 2005.

[20] Grant Schindler and Frank Dellaert. Probabilistic temporal inference on reconstructed 3D scenes. In *CVPR*, pages 1410–1417, 2010.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[22] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the World from Internet Photo Collections. *IJCV*, 80(2):189–210, 2007.

[23] Aparna Taneja, Luca Ballan, and Marc Pollefeys. Image based detection of geometric changes in urban environments. In *ICCV*, pages 2336–2343, 2011.

[24] Aparna Taneja, Luca Ballan, and Marc Pollefeys. City-Scale Change Detection in Cadastral 3D Models Using Images. In *CVPR*, pages 113–120, 2013.

[25] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *International Conference on Multimedia*, MM '10, pages 1469–1472. ACM, 2010.

[26] Jun Yang, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *International Workshop on Workshop on Multimedia Information Retrieval*, pages 197–206. ACM, 2007.

[27] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[28] Chenxi Zhang, Liang Wang, and Ruigang Yang. Semantic Segmentation of Urban Scenes Using Dense Depth Maps. In *ECCV*, pages 708–721, 2010.

[29] Guofeng Zhang, Jiaya Jia, Wei Xiong, Tien-Tsin Wong, Pheng-Ann Heng, and Hujun Bao. Moving Object Extraction with a Hand-held Camera. In *ICCV*, pages 1–8, 2007.

[30] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. *NIPS*, 2014.