Detecting Changes in 3D Structure of a Scene from Multi-view Images Captured by a Vehicle-mounted Camera

Ken Sakurada Takayuki Okatani Koichiro Deguchi
Tohoku University, Japan
{sakurada,okatani}@vision.is.tohoku.ac.jp

Abstract

This paper proposes a method for detecting temporal changes of the three-dimensional structure of an outdoor scene from its multi-view images captured at two separate times. For the images, we consider those captured by a camera mounted on a vehicle running in a city street. The method estimates scene structures probabilistically, not deterministically, and based on their estimates, it evaluates the probability of structural changes in the scene, where the inputs are the similarity of the local image patches among the multi-view images. The aim of the probabilistic treatment is to maximize the accuracy of change detection, behind which there is our conjecture that although it is difficult to estimate the scene structures deterministically, it should be easier to detect their changes. The proposed method is compared with the methods that use multi-view stereo (MVS) to reconstruct the scene structures of the two time points and then differentiate them to detect changes. The experimental results show that the proposed method outperforms such MVS-based methods.

1. Introduction

This paper considers a problem of detecting temporal changes in the three-dimensional structure of a scene, such as an urban area, from a pair of its multi-view images captured at two separate times. For the images, we consider those captured by a camera mounted on a ground vehicle while running it on city streets. The underlying motivation is to develop a method for automatically detecting the temporal changes of a whole city when it changes its structure in a relatively short time period because of disasters such as earthquakes and tsunamis. Its applications include quickly grasping the damages of a city caused by an earthquake by simply running a vehicle with a camera in the area (assuming its pre-earthquake images are also available) and visualizing the processes of short-time recovery or long-time reconstruction from them by similarly capturing images for multiple times.

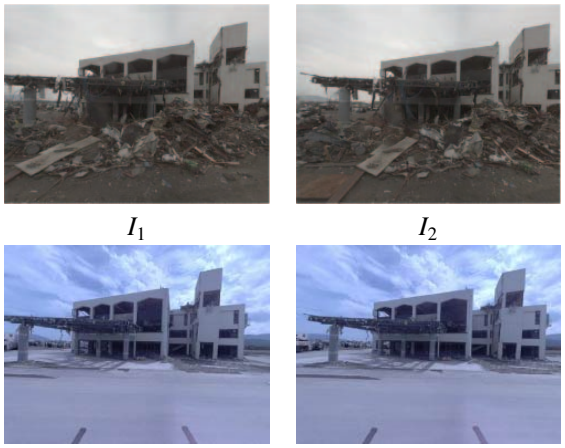


Figure 1. A pair of two images of the same scene taken at two separate times. (These are trimmed from omni-directional images.)

Exactly for the latter purpose, we are creating the image archives of the urban and residential areas damaged by the tsunami caused by the earthquake happened in Japan in March 2011. We have been periodically (every three to four months) capturing their images using a vehicle having an omni-directional camera on its roof. The target area is 500km long along the northern-east coastal line in Japan, and the image data accumulated so far amount to about 20 terabytes. Figure 1 shows examples of these images, which are a pair of two images of the same scene captured three months apart.

To achieve the goal of detecting temporal 3D scene changes from these images, a naive approach would be to use Multi-View Stereo (MVS) [5, 16] to reconstruct the 3D shapes of the area at different time points from their images and differentiate them to detect changes in 3D structure. Considering the recent success of MVS, this approach is seemingly promising. However, apart from the reconstruction from aerial imagery, which has achieved great success lately, it is still a difficult task to accurately reconstruct the structure of a scene from its images taken by a ground vehicle-mounted camera. Figure 2 shows the re-

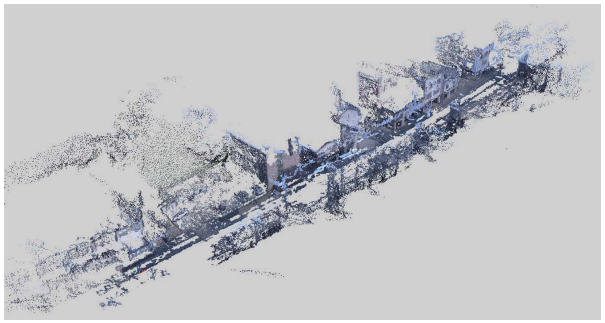


Figure 2. A result of applying PMVS2 [5] to our images that are obtained by a vehicle-mounted omni-directional camera at every few meters along a street. The camera poses needed for running PMVS2 are obtained by performing SfM.

sults of applying PMVS2, one of the state-of-the-art, to our images. It is observed from the results that there are a lot of missing parts in the reconstruction. (Some of the existing ones are also incorrectly reconstructed, although they cannot be judged from this picture alone.) These may be attributable to several reasons, such as the large depth variations which are contrasted with aerial imagery, the limited variety and number of camera poses (i.e., the viewpoints are on a straight line along the vehicle path), and the insufficient scene textures. The differentiation of the two reconstructions thus obtained does not give good results, as will be shown later.

In this paper, we propose another approach to this problem. The basic idea is that we want to know not the scene structure of each time point but their changes; thus, we formulate the problem so as to estimate them directly from the images. The core of the formulation, which distinguishes it from the above MVS-based one, is a probabilistic treatment of scene structures. To be specific, we estimate the scene structure (specifically, the scene depths from a selected viewpoint) not deterministically but probabilistically; namely, we obtain not a point estimate but a probabilistic density of depths; we then estimate whether the scene changes or not by integrating the obtained depth density in such a way that their ambiguity is well reflected in the final estimates. The overall estimation is performed in a probabilistic framework, where the inputs are the similarity of the local image patches among the multi-view images. The camera poses are necessary in this estimation and are estimated in advance by performing SfM for the images of each time point followed by registration of the reconstructions.

Our aim behind this probabilistic treatment of scene structures is to maximize the accuracy of detecting scene changes. If scene structure has to be deterministically determined even though observations give only ambiguous information, the two reconstructions will inevitably have errors, so do the estimated scene changes obtained by differentiating them. Our approach could reduce such errors by appropriately considering the ambiguity of scene structure. As a by-product, we can also reduce the computational time; it might be a waste to spend large computational resources to compute scene structures, as we need only their changes.

The paper is organised as follows. In Section 2, we summarize the related work. Section 3 explains how data are processed from image capture to change detection. In Section 4, we presents a novel algorithm for change detection. Section 5 shows several experimental results. Section 6 concludes this study.

2. Related work

Many researches have been conducted to develop methods for detecting temporal changes of a scene. However, most of them consider the detection of 2D changes (i.e., those only in image appearance), whereas we want to detect changes in 3D structure of scenes. Thus, there are only a limited number of studies that could potentially be applied to our problem.

The standard problem formulation of 2D change detection [12, 14] is such that an appearance model of a scene is learned using its n images and then based on $n + 1^{st}$ image, it is determined whether a significant change has occurred. Most of the studies of 3D change detection [3, 8, 7, 12, 18] follow a similar formulation; namely, a model of the scene in a "steady state" is built and a newly-captured image(s) is compared against it to detect changes.

In [12], targeting at aerial images capturing a ground scene, a method is proposed that learns a voxel-based appearance model of a 3D scene from its 20–40 images. Its improved method to minimize storage space is presented in [3]. In [8], a method is proposed that detects scene changes by estimating the appearance or disappearance of line segments in space. All of these studies create an appearance model of the target scene from a sufficiently large number of images. Such an approach is fit for aerial or satellite imagery or the case of stationary cameras, but is not fit for the images taken in our setting.

Several studies assume that a 3D model of the scene is given by using other sensors or methods than the images used for the change detection. In [7], assuming that the 3D model of a building is given, the edges extracted in its aerial images are matched with the projection of the 3D model to detect changes. The recent study of Taneja et al. [18] is of the same type. Their method detects temporal changes of a scene from its multi-view images, and thus it is close to ours from an application point of view. However, their motivation is to minimize the cost needed for updating the 3D model of a large urban area, and thus, a dense 3D model of the target scene is assumed to be given.

Our method differs from all of these in formulation of the problem. In our formulation, the changes of a scene are detected from two sets of images taken at two different time points. The two image sets are "symmetric" in the sense that they have similar sizes and are of the same nature. We

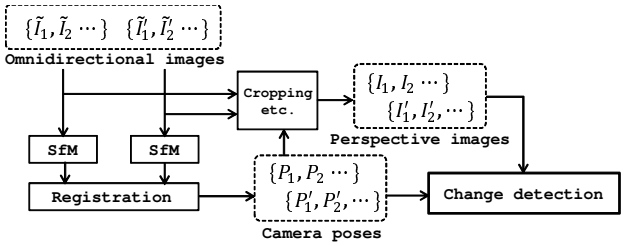
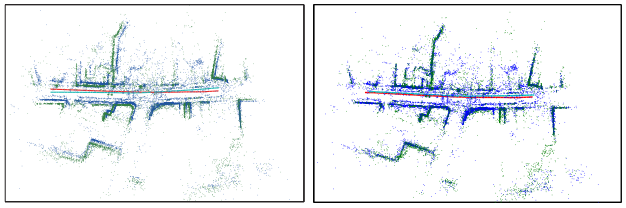


Figure 3. Data flow diagram; see texts for explanation.



(a) (b)

Figure 4. Registration of 3D reconstructions from two image sequences taken at different times. (a) Initial estimate. (b) Final result.

do not assume a dense 3D model of the scene to be given, or do not create one from the input images themselves, as it is difficult for the images captured from a ground vehicle-mounted camera; see Fig. 1. To do so, it is necessary to have a large number of multi-view images captured from a variety of viewpoints [1, 2, 13, 17, 22, 24], or to use a range sensor.

In the sense that the input data are symmetric, ours might be close to the study of Schindler and Dellaert [15]. They propose a method that uses a large number of images of a city that are taken over several decades to perform several types of temporal inferences, such as estimating when each building in the city was constructed. However, besides the necessity for a large number of images, their method represents scene changes only in the form of point clouds associated with image features.

3. From image acquisition to change detection

3.1. Image acquisition

As mentioned earlier, we have been periodically acquiring the images of the tsunami-devastated areas in the northern-east coast of Japan. The images are captured by a vehicle having an omni-directional camera (Ladybug3 of Point Grey Research Inc.) on its roof. An image is captured at about every 2m on each city street to minimize the total size of the data as well as to maintain the running speed of the vehicle under the constraint of the frame rate of the camera.

The goal of the present study is to detect the temporal changes of a scene from its images thus obtained at two separate times. Figure 3 shows how the input images are processed. For computational simplicity, our algorithm for change detection takes as inputs not the omni-directional images but the perspective images cropped from them. The algorithm also needs the relative camera poses of these images. To obtain them, we perform SfM for each sequence followed by registration of the two reconstructions, which are summarized below.

3.2. Estimation of relative camera poses

The algorithm shown in the next section uses only several perspective images to detect changes of a scene. For the reason of accuracy, however, to obtain their camera poses, we perform SfM and registration not with these perspective images alone but with a more number (e.g., 100 viewpoints) of omni-directional images that contain these viewpoints. To be specific, we do this in the following two steps. First, we perform SfM independently for each sequence. We employ a standard SfM method [6, 11, 21] with extensions to deal with omni-directional images [20]. Next, we register the two 3D reconstructions thus obtained as follows. We first roughly align the two reconstructions with a similarity transform; putative matches of the feature points are established between the two sequences based on their descriptor similarity, for which RANSAC is performed [4]. For the aligned reconstructions, we reestablish the correspondences of feature points by incorporating a distance constraint. Using the newly established correspondences along with original correspondences within each sequence, we perform bundle adjustment for the extended SfM problem, in which the sum of the reprojection errors for all the correspondences is minimized. Figure 4(a) shows the initial rough alignment of the two reconstructions and (b) shows the final result.

4. Detection of temporal changes of a scene

4.1. Problem

Applying the above methods to two sequences of omni-directional images, we have the camera pose of each image represented in the same 3D space. Choosing a portion of the scene for which we want to detect changes, we crop and warp the original images to have two sets of perspective images covering the scene portion just enough, as shown in Fig. 5. In this section, we consider the problem of detecting scene changes from these two sets of multi-view perspective images. For simplicity of explanation, we mainly consider the minimal case where there are two images in each set.

4.2. Outline of the proposed method

We denote the first set of images of time t by $\mathcal{I} = \{I_1, I_2\}$ and the second set of time t' by $\mathcal{I}' = \{I'_1, I'_2\}$. As shown in Fig. 6, one of the two image sets, $\mathcal{I}$, is used for estimating the depths of the scene, and the other image set $\mathcal{I}'$ is used

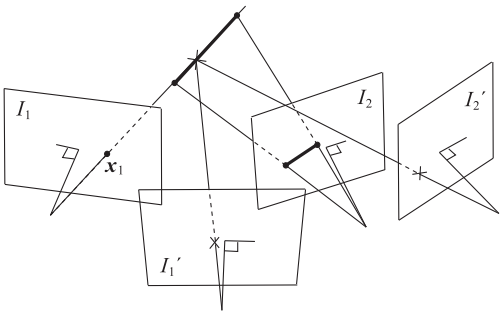


Figure 5. Geometry of two sets of multi-view perspective images taken at different times. For each pixel $\mathbf{x}_1$ of I_1, the probability that the scene depth has changed is estimated.

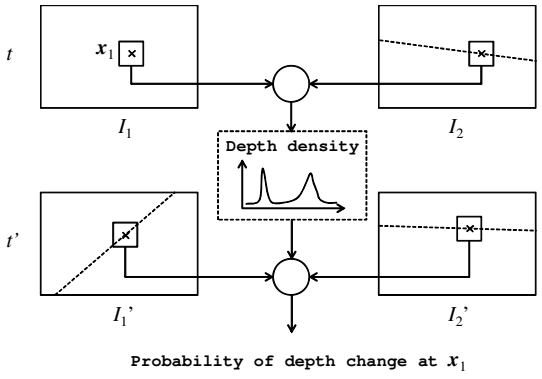


Figure 6. Outline of the proposed method. The probability density of the scene depth at a point $\mathbf{x}_1$ of I_1 is estimated from I_1 and I_2. This is combined with the comparison of the local patches of I_1' and I_2' to estimate the probability that the scene depth changes at $\mathbf{x}_1$ between t and t'. Note that the patches are compared only among the images taken at the same time. The broken lines in the images indicate epipolar lines associated with $\mathbf{x}_1$.

for estimating changes of the scene depths. (These may be swapped.) Choosing one image from $\mathcal{I}$, say I_1, which we call a *key frame* here, the proposed method considers the scene depth at each pixel of I_1 and estimates whether or not it changes from t to t'. The output of the method is the probability of a depth change at each pixel of I_1.

For the first image set $\mathcal{I}_1$, its images are used to estimate the depth map of the scene at t. To be specific, not the value of the depth d but its probabilistic density $p(d)$ is estimated. For the other set $\mathcal{I}'$, a spatial point having depth d at a certain pixel of the key frame I_1 is projected onto I_1' and I_2', respectively, as shown in Fig. 5, and then the similarity s_d' of the local patches around these two points is computed. The higher the similarity is, the more the spatial point is likely to belong to the surface of some object in the scene at t', and the inverse is true as well. The similarity s_d' is computed for each depth d, which gives a density function of d that is similar to $p(d)$.

By combining these two estimates, $p(d)$, and s_d', the proposed method calculates the probability of a depth change. In this process, the change probability evaluated for each depth d is integrated over d to yield the overall probability of a depth change. This makes it unnecessary to explicitly determine the scene depth neither at t nor t'. This is a central idea of the proposed method.

It should also be noted that our method evaluate the patch similarity only within each image set of $\mathcal{I}_1$ and $\mathcal{I}_2$. This makes it free from the illumination changes between the time points of the image capture.

4.3. Estimation of the density of scene depths

To estimate the density of scene depths, we use the similarity of local patches in the images, as is done in multi-view stereo [5, 9, 13, 16, 23]. By dividing the inverse depth in a certain range from near to far away into n discrete values, we denote the depth by indexes $d = 1, \ldots, n$. For a point $\mathbf{x}_1$ of I_1, we denote the projection onto I_2 of a spatial point lying on the ray of $\mathbf{x}_1$ and having depth d by $\mathbf{x}_2(d)$. The difference between the local patches around $\mathbf{x}_1$ and $\mathbf{x}_2(d)$ is evaluated by the similarity (rigorously dissimilarity) function

$$s_d(\mathbf{x}_1) = \frac{1}{3|\mathcal{W}|} \sum_{r,g,b} \sum_{\delta\mathbf{x}\in\mathcal{W}} \left| I_1(\mathbf{x}_1 + \delta\mathbf{x}) - I_2(\mathbf{x}_2(d) + \delta\mathbf{x}) \right|, \quad (1)$$

where $\mathcal{W}$ defines the size of the local patches. (We used 5×5 pixels in the experiment.)

Although s_d for correctly matched points will ideally be 0, it will not in practice because of image noise, shape changes of the patches, etc. Having examined s_d for correctly matched points, we found that its distribution is well approximated by a half Laplace distribution; see the supplementary note for details. Then, we model $p(s)$ as

$$p(d) \propto \exp(-s_d/\sigma), \quad (d = 1, \ldots, n). \quad (2)$$

The probabilities $[p(d = 1), \ldots, p(d = n)]$ are obtained by normalizing the above so that their sum will be 1. We set $\sigma = 1.5$ in the experiments based on the statistics of real images; see the supplementary note.

4.4. Estimating probabilities of scene changes

We introduce a binary variable c to represent whether or not the scene depth at a pixel $\mathbf{x}_1$ of the key frame I_1 has changed from t to t'; $c = 1$ indicates it has changed and $c = 0$ it has not.

Suppose projecting onto I_1' and I_2' a spatial point lying on the ray of $\mathbf{x}_1$ and having depth d, as shown in Fig. 5. We denote these two points by $\mathbf{x}_1'(d)$ and $\mathbf{x}_2'(d)$, respectively. Similarly to Eq. (1), the difference of the local patches around these two points is calculated as

$$s_d' = \frac{1}{3|\mathcal{W}|} \sum_{r,g,b} \sum_{\delta\mathbf{x}\in\mathcal{W}} \left| I_1'(\mathbf{x}_1'(d) + \delta\mathbf{x}) - I_2'(\mathbf{x}_2'(d) + \delta\mathbf{x}) \right| \quad (3)$$

Computing $s'_1, ..., s'_n$ for the depths $d = 1, ..., n$ from the images, we consider evaluating the following posterior probability given $s'_1, ..., s'_n$ as observations:

$$p(c = 1|s'_1, ..., s'_n). \tag{4}$$

This directly gives the probability that the scene changes its structure at the pixel $\mathbf{x}_1$ of I_1. This can be rewritten by Bayes' rule as

$$p(c = 1|s'_1, ..., s'_n) = \frac{p(s'_1, ..., s'_n|c = 1)p(c = 1)}{p(s'_1, ..., s'_n)}. \tag{5}$$

The denominator is given by

$$\begin{aligned} p(s'_1, ..., s'_n) = p(s'_1, ..., s'_n|c = 1)p(c = 1) \\ + p(s'_1, ..., s'_n|c = 0)p(c = 0). \end{aligned} \tag{6}$$

Here, the term $p(c = 1)$ is the prior probability that the scene depth changes at this pixel. We set a constant number to $p(c = 1)$. Its inverse $p(c = 0)$ is given by $p(c = 0) = 1 - p(c = 1)$.

We next evaluate $p(s'_1, ..., s'_n|c = 1)$ and $p(s'_1, ..., s'_n|c = 0)$. We assume that $s'_d(d = 1, \ldots, n)$ is independent of each other and that

$$p(s'_1, ..., s'_n|c = 1) = \prod_{d=1}^{n} p(s'_d|c = 1), \tag{7a}$$

$$p(s'_1, ..., s'_n|c = 0) = \prod_{d=1}^{n} p(s'_d|c = 0). \tag{7b}$$

To further analyze $p(s'_d|c = 1)$ and $p(s'_d|c = 0)$, we introduce a binary variable δ_d to represent whether or not the scene depth (at $\mathbf{x}_1$ of I_1 at time t) is d, that is, whether or not the spatial point having depth d belongs to the surface of some object at t; $\delta_d = 1$ indicates this is the case and $\delta_d = 0$ otherwise. Using δ_d, $p(s'_d|c = 1)$ can be decomposed as follows:

$$\begin{aligned} p(s'_d|c = 1) &= p(s'_d, \delta_d = 1|c = 1) + p(s'_d, \delta_d = 0|c = 1) \\ &= p(s'_d|\delta_d = 1, c = 1)p(\delta_d = 1) \\ &\quad + p(s'_d|\delta_d = 0, c = 1)p(\delta_d = 0), \end{aligned} \tag{8}$$

where $p(\delta_d = 1|c = 1) = p(\delta_d = 1)$ and $p(\delta_d = 0|c = 1) = p(\delta_d = 0)$ are used, which is given by the independence of δ_d and c. The density $p(s'_d|c = 0)$ can be decomposed in a similar way. The term $p(\delta_d = 1)$ in Eq. (8) is the probability that the scene depth is d, and thus it is equivalent to $p(d)$ that has been already obtained; thus, $p(\delta_d = 1) = p(d)$. The term $p(\delta_d = 0)$ is given by $p(\delta_d = 0) = 1 - p(\delta_d = 1) = 1 - p(d)$.

To evaluate Eq. (8), we need to further consider the conditional densities $p(s'_d|\delta_d = 1, c = 1)$ and $p(s'_d|\delta_d = 0, c = 1)$. There are four combinations of (δ_d, c): $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$. Each combination can be related to whether the scene depth is d at time t' or not. For example, $(\delta_d, c) = (1, 0)$ means that the scene depth is d at time t and remains so at t'; $(\delta_d, c) = (1, 1)$ means that the scene depth is d at t and is not so at t'. Let δ'_d be a binary variable indicating whether or not the scene depth is d at time t'; $\delta'_d = 1$ if the scene depth is d at t' and $\delta'_d = 0$ otherwise. Table 1 shows the values of δ'_d for all the combinations. Note that the combination $(\delta_d, c) = (0, 1)$, which means that the scene depth is not d at t and changes at t', does not fully constrain δ'_d. Thus we denote it by δ'_d is either 0 or 1.

Table 1. Values of δ'_d for different pairs of c and δ_d. The definition of the variables is as follows: $c = 1$ indicates the scene depth changes from t to t' and $c = 0$ otherwise; $\delta_d = 1$ indicates the scene depth is d at time t and $\delta_d = 0$ otherwise; δ'_d is the same as δ_d but not at t but t'.

c \ δ_d	0	1
0	0	1
1	0 or 1	0

From the table, we can rewrite the conditional densities for the four combinations as

$$p(s'_d|\delta_d = 0, c = 0) = p(s'_d|\delta'_d = 0), \tag{9a}$$

$$p(s'_d|\delta_d = 0, c = 1) = p(s'_d|\delta'_d = 0 \text{ or } 1), \tag{9b}$$

$$p(s'_d|\delta_d = 1, c = 0) = p(s'_d|\delta'_d = 1), \tag{9c}$$

$$p(s'_d|\delta_d = 1, c = 1) = p(s'_d|\delta'_d = 0). \tag{9d}$$

The densities on the right hand side can be modelled as follows. When $\delta'_d = 0$, which means the scene depth is not d (at t'), s'_d measures the similarity between the patches of two different scene points. Thus, we model $p(s'_d|\delta'_d = 0)$ by a uniform distribution and set

$$p(s'_d|\delta'_d = 0) = \text{const.} \tag{10}$$

When $\delta'_d = 1$, on the other hand, s'_d measures the similarity between the patches of the same scene point. Then, this is exactly the same situation as s_d for correctly matched points. Thus, using the same half Laplace distribution as s_d, we set $p(s'_d|\delta'_d = 1) \propto \exp\left(-s'_d/\sigma'\right)$. In the experiments, we set $\sigma'(= \sigma) = 1.5$.

The conditional density $p(s'_d|\delta_d = 0, c = 1)$ can be factorized as follows:

$$\begin{aligned} &p(s'_d|\delta_d = 0, c = 1) \\ &\quad = p(s'_d|\delta'_d = 0, \delta_d = 0, c = 1)p(\delta'_d = 0|\delta_d = 0, c = 1) \\ &\quad + p(s'_d|\delta'_d = 1, \delta_d = 0, c = 1)p(\delta'_d = 1|\delta_d = 0, c = 1). \end{aligned} \tag{11}$$

The probability $p(\delta'_d = 1|\delta_d = 0, c = 1)$ is difficult to quantify, but, fortunately, it should be small. Thus, we approximate $p(s'_d|\delta_d = 0, c = 1) \approx p(s'_d|\delta'_d = 0, \delta_d = 0, c = 1)$.

Using the derived equations and the introduced models,

the conditional probability $p(c = 1|s'_1, \ldots, s'_n)$ can be evaluated for each $\mathbf{x}_1$. We may judge that if the probability is higher than 0.5, the scene depth has changed at the pixel, and it has not changed, otherwise.

We have considered the minimal case of using a pair of images for each time. When two or more pairs of images are available, we can use them to improve estimation accuracy. In the experiments, we use a naive method, which integrates the observations from the multiple image pairs based on an assumption that they are independent of each other.

5. Experimental results

We conducted several experiments to examine the performance of the proposed method. For the experiments, we chose a few scenes and their images from our archives mentioned in Sec.3.1. The chosen images are taken at one and four months after the tsunami[1]. Typically, a lot of tsunami debris appear in the earlier images, whereas they disappear in the later ones because of recovery operations. We wish to correctly identify their disappearance in the later images.

The proposed method uses two or more images for each time. In the experiment, we use four images of consecutive viewpoints for each time, i.e., three pairs of images. These are perspective images (cropped from omni-directional images) of 640×480 pixel size. The disparity space is discretized into 128 blocks ($n = 128$). Assuming that there is no prior on the probability of scene changes, we set $p(c = 1) = 0.5$. It is noted, though, that in the experiments, the results are very robust to the choice of this value; see the supplementary note for details. These are fixed for all the experiments.

5.1. Compared methods

We compared our method with MVS-based ones, which first reconstruct the structures of a scene based on MVS and differentiate them to obtain scene changes. We consider two MVS algorithms for 3D reconstruction, PMVS2 [5] and a standard stereo matching algorithm for it.

In the former case, PMVS2 is applied to a sufficiently long sequence of images (e.g., 100 viewpoints) covering the target scene. Our omni-directional camera consists of six cameras and records six perspective images at each viewpoint. All these six images per viewpoint are inputted to PMVS2 after distortion correction. PMVS2 outputs point clouds, from which we create a depth map viewed from the key frame. This is done by projecting the points onto the image plane in such a way that each point occupies an image area of 7×7 pixels. Two depth maps are created for the two time points and are differentiated to obtain scene changes. We call the overall procedure PMVS2.

[1]The data used in this study (the omni-directional image sequences of the chosen streets and our estimates of their camera poses) are available from our web site: http://www.vision.is.tohoku.ac.jp/us/download/.

In the latter case, a standard stereo matching algorithm is used, in which a MRF model is assumed that is defined on the four-connected grid graph; the local image similarity is used for the data term and a truncated l_1 norm $f_{ij} = \max(|d_i - d_j|, d_{\max}/10)$ is used for the smoothness term. We use two types of similarity; one is the SAD-based one (Eq. (1) and Eq. (3)) that is used in our method, and the other is the distance between SIFT descriptors at the corresponding points [19]. Then, the optimization of the resulting MRF models is performed using graph cuts [10]. Similarly to the above, two depth maps are computed and are differentiated to obtain scene changes. We call these procedures patch-MVS and SIFT-MVS.

5.2. Comparison of the results

Figure 7 shows the results for a scene. From left to right columns, the input images with a hand-marked ground truth, the results of the proposed method, PMVS2, Patch-MVS, and SIFT-MVS, respectively. For the proposed method, besides the detected changes, the change probability $p(c = 1|\cdots)$ is shown as a grey-scale image; its binarized version by a threshold $p > 0.5$ gives the result of change detection. For each of the MVS-based methods, besides the result, two estimated depths maps for the different times are shown. The detection result is their differences. Whether the scene changes or not is judged by whether the difference in its disparity is greater than a threshold. We chose 6 (disparity ranges in $[0 : 127]$) for the threshold, as it achieves the best results in the experiments. The red patches in the depth maps of PMVS2 indicate that there is no reconstructed point in the space.

Comparing the result of the proposed method with the ground truth, it is seen that the proposed method can correctly detect the scene changes, i.e., the disappearance of the debris and the digger; the shape of the digger arm is extracted very accurately. There are also some differences. The proposed method cannot detect the disappearance of the building behind the digger and of the thin layer of sands on the ground surface. The former is considered to be because the building is occluded by the digger in other viewpoints. The proposed method does not have a mechanism of explicitly dealing with occlusions but using multiple pairs of images, which will inevitably yield some errors. For the layer of sands, its structural difference might be too small for the proposed method to detect it.

The results of the MVS-based methods are all less accurate than the proposed method. As these methods differentiate the two depth maps, a slight reconstruction error in each will results in a false positive. Thus, even though their estimated depths appear to capture the scene structure mostly well, the estimated scene changes tends to be worse than the impression we have for each depth map alone.

There are in general several causes of errors in MVS-based depth estimation. For example, MVS is vulnerable

to objects without textures (e.g., the ground surface in this scene). PMVS2 does not reconstruct objects that do not have reliable observations, e.g., textureless objects. As the proposed method similarly obtains depth information from image similarity, the same difficulties will have bad influence on the proposed method. However, it will be minimized by the probabilistic treatment of the depth map; taking all probabilities into account, the proposed method makes a binary decision as to whether a scene point changes or not.

We obtain precision and recall for each result using the ground truth and then calculate its F_1 score; it is 0.76, 0.59, 0.53, 0.71, in the order of Fig. 7, respectively.

Figure 8 shows results for other images. From top to bottom rows, I', the ground truths, the results of the proposed method, and those of SIFT-MVS are shown, respectively. It is seen that the proposed method produces better results for all the images. This is quantitatively confirmed by their F_1 scores which are shown in Table 2.

6. Conclusions

We have described a method for detecting temporal changes of the 3D structure of an outdoor scene from its multi-view images taken at two separate times. These images are captured by a vehicle-mounted camera running in a city street. The method estimates the scene depth probabilistically, not deterministically, and judges whether or not the scene depth changes in such a way that the ambiguity of the estimated scene depth is well reflected in the final estimates. We have shown several experimental results, in which the proposed method is compared with MVS-based methods, which use MVS to reconstruct the scene structures and differentiate two reconstructions to detect changes. The experimental results show that the proposed method outperforms the MVS-based ones.

It should be noted that our method estimates scene changes independently at each image pixel; no prior on the smoothness or continuity of scene structures is used. This is contrasted with MVS, which always uses some prior about them. Such priors, which have been confirmed to be very effective in dense reconstruction, are in reality a double-edged sword. We may say that the reason why MVS needs such priors is because it has to deterministically determine scene structures even if only insufficient observations are available. Considering that our method achieves better results (even) without such priors, it could be possible that such priors do more harm than good as far as change detection is concerned.

Acknowledgement

This work was supported by Grant-in-Aid for Scientific Research on Innovative Areas "Shitsukan" (No. 23135501) and JSPS KAKENHI Grant Number 2230057.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV*, pages 72–79, 2009.

[2] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-Continuous Optimization for Large-Scale Structure from Motion. In *CVPR*, pages 3001–3008, 2011.

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

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[5] Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. *PAMI*, 32(8):1362–1376, 2010.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision Second Edition*. Cambridge University Press, 2004.

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

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

[9] S. B. Kang, R. Szeliski, and J. Chai. Handling Occlusions in Dense Multi-view Stereo. In *CVPR*, pages I–103–I–110, 2001.

[10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–59, 2004.

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

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

[13] 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.

[14] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image Change Detection Algorithms: A Systematic Survey. *TRANSACTIONS ON IMAGE PROCESSING*, 14(3):294–307, 2005.

[15] G. Schindler and F. Dellaert. Probabilistic temporal inference on reconstructed 3D scenes. In *CVPR*, pages 1410–1417, 2010.

[16] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *CVPR*, pages 519–528, 2006.

[17] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *IJCV*, 80(2):189–210, 2007.

[18] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *ICCV*, pages 2336–2343, 2011.

[19] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *PAMI*, 32(5):815–830, 2010.

[20] A. Torii, M. Havlena, and T. Pajdla. From Google Street View to 3D City Models. In *ICCV Workshops*, pages 2188–2195, 2009.

[21] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment - Modern Synthesis. In *ICCV*, pages 298–372, 1999.

[22] C. Zhang, L. Wang, and R. Yang. Semantic Segmentation of Urban Scenes Using Dense Depth Maps. In *ECCV*, pages 708–721, 2010.

[23] G. Zhang, J. Jia, T.-t. Wong, and H. Bao. Recovering Consistent Video Depth Maps via Bundle Optimization. In *CVPR*, pages 1–8, 2008.

[24] G. Zhang, J. Jia, W. Xiong, T.-T. Wong, P.-A. Heng, and H. Bao. Moving Object Extraction with a Hand-held Camera. In *ICCV*, pages 1–8, 2007.

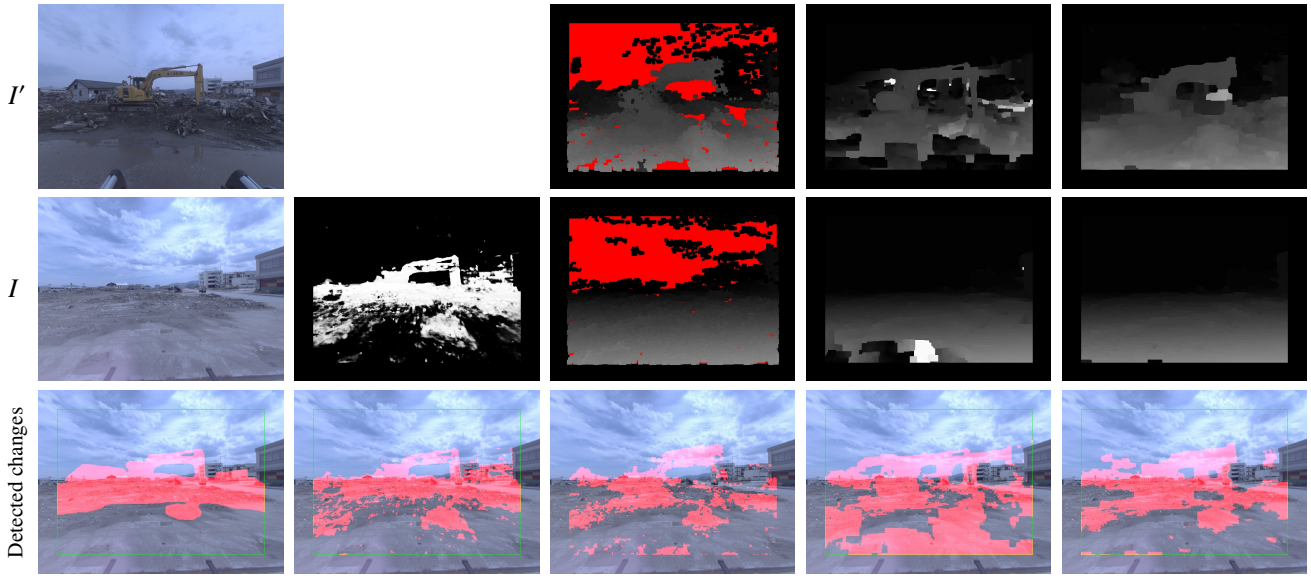


Figure 7. Results of the proposed method and the three MVS-based ones for a scene. From left to right columns, the input images and the ground truth, the results of the proposed methods, and those of PMVS2, Patch-MVS, and SIFT-MVS, respectively. The third row shows the detected changes. The numbers in their captions are the F_1 scores representing accuracy of the detection.

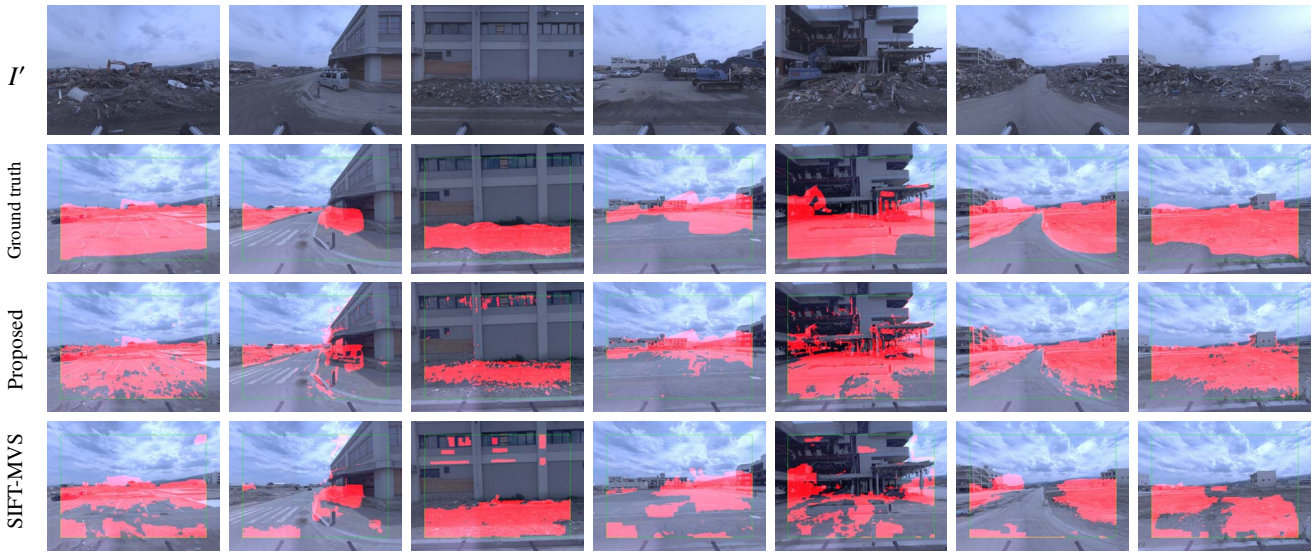


Figure 8. Results for other images. From top to bottom rows, I', hand-marked ground truths, results of the proposed method, and those of SIFT-MVS.

Table 2. F_1 scores of the detected changes shown in Fig. 8.

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	Average
Proposed	0.88	0.67	0.77	0.85	0.82	0.91	0.92	0.83
PMVS2	0.49	0.30	0.65	0.66	0.56	0.58	0.66	0.56
Patch-MVS	0.66	0.28	0.69	0.60	0.70	0.65	0.77	0.62
SIFT-MVS	0.68	0.41	0.73	0.71	0.60	0.67	0.73	0.65