Diminished Reality Considering Background Structures

Norihiko Kawai Tomokazu Sato Naokazu Yokoya*

Graduate School of Information Science, Nara Institute of Science and Technology, Japan

Abstract

This paper proposes a new diminished reality method for 3D scenes considering background structures. Most conventional methods using image inpainting assumes that the background around a target object is almost planar. In this study, approximating the background structure by the combination of local planes, perspective distortion of texture is corrected and searching area is limited for improving the quality of image inpainting. The temporal coherence of texture is preserved using the estimated structures and camera pose estimated by Visual-SLAM.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.4.9 [Computing Methodologies]: Image Processing and Computer Vision—Applications

1 Introduction

Diminished reality which visually removes real objects by replacing them with background textures in video images in real time can be used for various applications. For example, some pieces of furniture are removed for arrangement simulation, AR markers are hidden for achieving seamless fusion between virtual objects and the real world, and so on. Among diminished reality methods, for the scenes in which actual backgrounds of target objects cannot be observed or for the cases where it is burdensome for users to capture the backgrounds, image inpainting techniques, which generate plausible textures using information around the target objects, has been often used [2, 3, 4]. In this study, we focus on the image inpainting-based approach.

Herling et al. [2, 3] have applied an exemplar-based image inpainting method to an original input image. Although the exemplar-based methods yield good results in many cases, it is known that they are weak for perspective distortion of regular patterns. To solve this problem, our previous method [4] has corrected the perspective distortion using an AR marker so that the size of regular texture patterns can be unified. In the proposed method, we apply this idea by fitting multiple planes to feature points estimated by Visual-SLAM and rectifying an input image. In addition, although conventional methods [2, 3, 4] search the whole image for similar patterns, we add a constraint for automatically limiting searching regions using structures around a target object to improve the image quality. On the other hand, as for temporal consistency, Herling et al. [3] used a homography for determining searching areas in the next frame on the assumption that the background is almost planar. Our previous method [4] has also used a homography for synthesizing an inpainted result for hiding an AR marker. As a result, these methods successfully have preserved temporal coherence for planar scenes. However, if the target scene is not planar, changes in appearances of textures on different structures cannot be compensated for using one unique homography. In the proposed method, the scene around a target object is divided into multiple planes whose number is automatically determined, and inpainted textures are successfully overlaid on the target object under comparatively unrestricted camera motion using the estimated multiple planes and camera pose by Visual-SLAM.

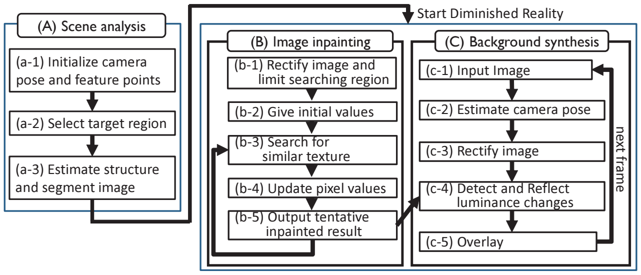
Start Diminished Reality
(A) Scene analysis
(a-1) Initialize camera pose and feature points
(a-2) Select target region
(a-3) Estimate structure and segment image
(B) Image inpainting
(b-1) Rectify image and limit searching region
(b-2) Give initial values
(b-3) Search for similar texture
(b-4) Update pixel values
(b-5) Output tentative inpainted result
(C) Background synthesis
(c-1) Input Image
(c-2) Estimate camera pose
(c-3) Rectify image
(c-4) Detect and Reflect luminance changes
(c-5) Overlay
next frame

Figure 1: Flow of the proposed method.

2 Proposed Diminished Reality

As shown in Figure 1, the proposed method first analyzes the target scene (A). Diminished reality is then achieved by a semi-dynamic approach which conducts inpainting for a key frame and synthesis for every frame concurrently like our previous method [4]. Although process (B) is not performed in real-time for generating high-quality texture, users can start applications without waiting time by performing the processes (B) and (C) concurrently. In the following, we describe the processes in detail.

2.1 Scene Analysis

As a pre-processing of diminished reality, the target scene is analyzed and the image is segmented into multiple regions. Concretely, after initializing the camera pose and feature points by Visual-SLAM, users manually specify a target region. The image coordinates of pixels in the specified target region are projected onto background planes, which are generated in the process described below, and the regions are fixed through all frames on the assumption that target objects exist almost on the background objects. Next, feature points in a certain range from the target are picked up and connected using Delaunay triangulation, and normal vectors of feature points are determined using the generated triangles.

Next, each feature point is classified into multiple groups based on the difference between the normal vector of feature point and the mean normal vector of feature points in a group, which is iteratively updated using mean-shift [1]. It should be noted that the number of groups are automatically determined. Concretely, first, a feature point is randomly selected as $\mathbf{x}_i$ (i is an index of a feature point) and normal vector $\mathbf{m}$ is calculated in an iterative manner as follows:

$$\mathbf{m}_t(\mathbf{n}_i) = \frac{\Sigma_{j=1}^{N} w(\mathbf{n}_j)\mathbf{n}_j}{M}, \tag{1}$$

$$w(\mathbf{n}_j) = \begin{cases} 1 & \left(\mathbf{n}_j \cdot \mathbf{m}_{t-1}(\mathbf{n}_i) > C\right) \\ 0 & (otherwise), \end{cases} \tag{2}$$

where $\mathbf{n}_i$ and $\mathbf{n}_j$ indicate unit normal vectors of feature point $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbf{m}_t(\mathbf{n}_i)$ means an unit mean vector in the t-th iteration when starting the process with $\mathbf{m}_0(\mathbf{n}_i) = \mathbf{n}_i$. M is a normalization factor. N is the number of feature points picked up in the above process and

*e-mail:{norihi-k, tomoka-s, yokoya}@is.naist.jp

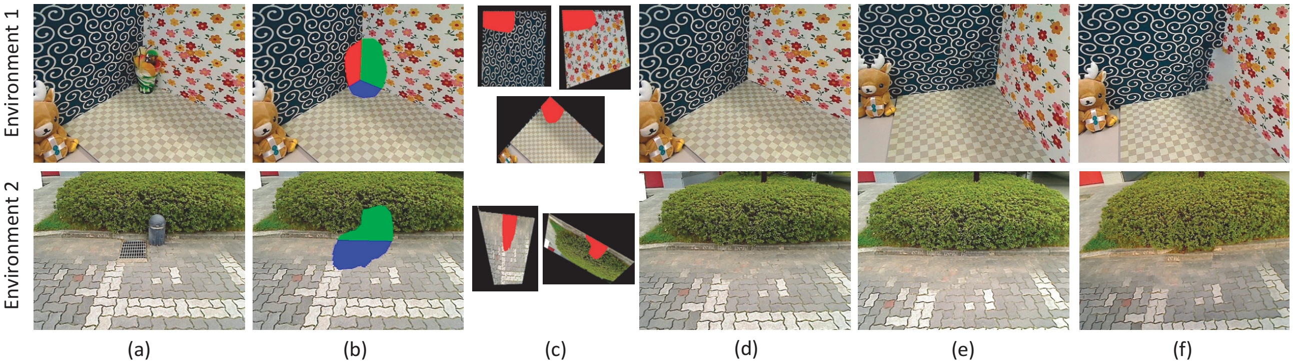

Figure 2: Results for two environments: (a) input images, (b) segmented target regions based on estimated planes, (c) rectified image for each plane, (d) results of proposed method from a viewpoint almost the same as the key frame, (e) results of proposed method from a different viewpoint and (f) results of a conventional approach using one homography.

C is a constant threshold. After several iterations, feature points $\mathbf{n}_j$ which satisfy $\mathbf{n}_j \cdot \mathbf{m}_t(\mathbf{n}_i) > C$ are labeled as the same group and these points are removed. Among the remaining feature points, a new feature point is randomly selected and the above processes are repeated until all feature points are labeled.

Finally, a plane is fitted on feature points of each group using LMeds (The Least Median of Squares). Here, the number of planes is basically the same as that of labeled groups, but if the number of feature point in a group is much smaller than the others, the group is removed. Then considering the depth from the camera to each plane, the whole image including the target region is segmented.

2.2 Image Inpainting based on Multiple Planes

In the image inpainting process, utilizing the fitted planes and segmented regions, the missing region is filled in by an exemplar-based image inpainting method. Here, an arbitrary exemplar-based method is adaptable in the proposed framework. As the concrete process, perspective distortion of an input image is corrected by calculating a homography matrix for each plane. The number of rectified images is the same as that of planes. Next, searching regions in image inpainting are limited based on the segmented image so that textures on the other planes cannot be used as exemplars for inpainting. We then apply an exemplar-based image inpainting method to the each generated image. Finally, inpainted results are combined using inverse homography matrices. Here, a frame which is used for inpainting is set as a key frame.

2.3 Real-time Overlay of Inpainted Textures

In process (C), the luminance of the inpainted results generated by image inpainting process (B) are adjusted based on difference in the luminance between the current frame and the key frame, as our previous method [4], and overlaid onto the target region based on the position of each plane and a camera pose estimated by Visual-SLAM for every frame. For luminance adjustment, we compare the intensities of the region around the target region in the rectified images between the key frame and a current frame. We then estimate luminance change ratios for all pixels in the target region by energy minimization. All pixel values in the inpainted result are then multiplied by the change ratios. Here, unlike our previous method [4], the luminance change ratios for the inpainted result of each plane image are separately calculated because the degree of luminance changes on each plane is thought to be different due to the different normal direction of each plane.

3 Experiments

We performed experiments using a PC with Core i7-3820QM 2.7 GHz CPU, 8 GB of memory, and GeForce GT 650M GPU for input images with resolution 640 × 480 captured by a USB camera (Logicool Qcam Pro 9000). We used the GPU for image rectification. We used PTAM [6] as Visual-SLAM and our previous method [5] as an inpainting method. We set threshold $C = cos(\pi/6)$ for segmentation.

Figure 2 shows results and comparison in two environments. The proposed method successfully segmented and rectified the images by fitting planes as in Figures (b) and (c) and gave natural textures to the target regions in the images, which have various types of textures such as regular and random patterns, from different viewpoints as in Figures (d) and (e). On the other hand, edges are made disconnected on the boundaries of the target regions by the conventional approach using one homography as in Figure (f) because the target region does not consist of one plane. In the environment 1, the proposed method worked at about 21 fps. As the limitation of the proposed framework, the quality of temporal coherence of textures largely depends on the robustness of Visual-SLAM. In the experiments, because we used PTAM [6] as a Visual-SLAM method, target scenes were required to have some textures.

4 Conclusion

We have proposed a new diminished reality method for 3D scenes. By approximating the background structure by the combination of local planes, perspective distortion of texture is corrected and searching area is limited for improving the quality of image inpainting. The temporal coherence of texture is preserved using the estimated structures and camera pose. In future work, we will develop a diminished reality method for scenes with more complex backgrounds.

Acknowledgements

This work was partially supported by JSPS KAKENHI Nos. 23240024 and 24700118.

References

[1] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1):32–40, 1975.

[2] J. Herling and W. Broll. Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments. In *Proc. ISMAR*, pages 207–212, 2010.

[3] J. Herling and W. Broll. PixMix: A real-time approach to high-quality diminished reality. In *Proc. ISMAR*, pages 141–150, 2012.

[4] N. Kawai, M. Yamasaki, T. Sato, and N. Yokoya. AR marker hiding based on image inpainting and reflection of illumination changes. In *Proc. ISMAR*, pages 293–294, 2012.

[5] N. Kawai and N. Yokoya. Image inpainting considering symmetric patterns. In *Proc. ICPR*, pages 2744–2747, 2012.

[6] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. ISMAR*, pages 225–234, 2007.