Multimed Tools Appl (2017) 76:9443–9462
DOI 10.1007/s11042-016-3550-8

Image and video completion via feature reduction and compensation

Mariko Isogawa[1] · Dan Mikami[1] · Kosuke Takahashi[1] · Akira Kojima[1]

Received: 8 November 2015 / Revised: 1 April 2016 / Accepted: 19 April 2016 /
Published online: 10 May 2016

Abstract This paper proposes a novel framework for image and video completion that removes and restores unwanted regions inside them. Most existing works fail to carry out the completion processing when similar regions do not exist in undamaged regions. To overcome this, our approach creates similar regions by projecting a low dimensional space from the original space. The approach comprises three stages. First, input images/videos are converted to a lower dimensional feature space. Second, a damaged region is restored in the converted feature space. Finally, inverse conversion is performed from the lower dimensional space to the original space. This generates two advantages: (1) it enhances the possibility of applying patches dissimilar to those in the original color space and (2) it enables the use of many existing restoration methods, each having various advantages, because the feature space for retrieving the similar patches is the only extension. The framework's effectiveness was verified in experiments using various methods, the feature space for restoration in the second stage, and inverse conversion methods.

Keywords Completion · Inpainting · Restoration · Low-dimensional feature space · Image transfer

1 Introduction

Photos and videos sometimes include unwanted objects such as a person walking in front of a filming target or a trash can on a beautiful beach. In this paper we call areas

Electronic supplementary material The online version of this article (doi:10.1007/s11042-016-3550-8) contains supplementary material, which is available to authorized users.

✉ Mariko Isogawa
isogawa.mariko@lab.ntt.co.jp

[1] NTT Media Intelligence Laboratories, 1-1 Hikarinooka, Yokosuka, Japan

containing such unwanted objects "damaged regions". Completion, also known as inpainting, is a method that deletes such areas[1]. It is acknowledged as one of the most important topics in many research fields, including augmented reality (AR), mixed reality (MR), and image processing [6, 11].

The most primitive solution for this problem restores damaged regions on a pixel-by-pixel basis with neighboring pixels. Bertalmio et al. restore damaged regions by propagating pixel values from surrounding pixels along with the brightness gradient [1], assuming that the smooth changes in pixel values within the border area enable natural image restoration. Though the method keeps luminance continuity with neighboring pixels, it still has difficulty in maintaining temporal and structural consistency. They also proposed an interesting extension [2]. It divides a target image to be restored into high and low frequency images; the low frequency image is filled by [1], while the high frequency image is restored by texture synthesis. The two restored images are then combined to make a final restored image. Although this extension is effective for restoring images with occasional or uniform texture, it has difficulties in restoring images with a complicated structure or a large damaged region.

The patch-based method, which aims at maintaining consistency well even for large damaged regions, is acknowledged as a promising approach. The method first selects a target patch to be restored that includes both source and damaged regions. Then it retrieves a similar patch to the target patch from the source region. Finally, the damaged region within the target patch is filled by using the obtained similar patch. The way similar patches are retrieved is one of the most important aspects for restoration quality.

Since the color-based patch retrieval [3] was proposed, edges [15] or motions [17] have been added to obtain more appropriate patches. With restrictive constraints, restoration works well if there are patches that have satisfactory matches for all features. In other words, these approaches implicitly assume that the target image includes such patches. However, this assumption does not hold when the region to be filled in contains complex structures or color distribution. Therefore, obtaining good results becomes difficult when the target patches include complicated shapes and/or have vast possible value spaces due to the lack of an appropriate patch for completion.

To overcome such insufficiency of patches, some previous methods used geometric deformations and changes in illumination [4, 7, 11]. The concept of these methods can be summarized as enhancing the availability of patches by transforming patches that are unsuitable in their original condition. However, these methods require huge computational cost for patch retrieval. Another method was developed by Shiratori et al., who proposed a technique for restoring video in a motion feature space [16]. Since this method uses motion features only for patch retrieval, it can be considered that it relaxes patch retrieval criteria. We think this method is quite important because it enables the restoration to be carried out not in the original feature space but in the converted feature space. This method is discussed in more detail in Section 2.1.

We propose a general framework for completing image and video (we use "content" to represent "image and video" when we do not need to distinguish them) via restoration in a different feature space from the original. The feature space for restoration we use is a lower dimensional feature space. This enables dissimilar patches in the original feature space to become similar in the lower dimensional space. For example, different colors in RGB space

[1] Although it is not a unified definition, the word "completion" tends to be used in cases where a missing region to be filled in becomes large [14].

$(R_1, G_1, B_1) = (200, 90, 126)$ and $(R_2, G_2, B_2) = (75, 156, 114)$ become the same gray scale value (127) when they are projected to gray scale. That is, conversion to a lower dimensional space relaxes "similarity" of patches. The framework consists of three stages (Fig. 1b): (1) converting input content to a lower dimensional feature space, (2) restoring the content in the converted lower dimensional feature space, and (3) inversely converting the restored content from the lower dimensional feature space to the original feature space.

The remainder of this paper is organized as follows. In Section 2 we briefly review related work. We describe the new framework we propose in Section 3 and in Section 4 show how it works by observing image and video completion results. In Section 5 we show that it also works well with various feature spaces, and with some state-of-the-art completion methods. In Section 6, we discuss the framework's current limitations and further studies. Finally, we conclude in Section 7 with a brief summary.

2 Related work

This section reviews previous studies for content restoration and feature creation. In Section 2.1 we describe restoration methods that increase patch availability. In Section 2.2 we review methods to create and add features to the content.

2.1 Approaches for increasing the availability of patches

The process most patch-based methods use is as follows: (1) choose a target patch P_t to be restored, (2) retrieve a similar patch P_s that maximizes $S(P_t, P_s)$, where S is a similarity function, (3) use P_s as a basis for restoring the damaged region within P_t. In patch-based completion studies, various methods to increase the availability of patches have been proposed. Here, we briefly introduce existing studies in two concept categories.

The first one is to increase the patch availability by transforming patches that are unsuitable in their original condition. Darabi et al. [4] introduce scaling and location while Huang et al. [7] allow projective transformation; both have reported good results. Kawai et al. use patches under different illumination [11]. These methods can be implemented by allowing patch deformations or illumination changes in process step (2). However, because

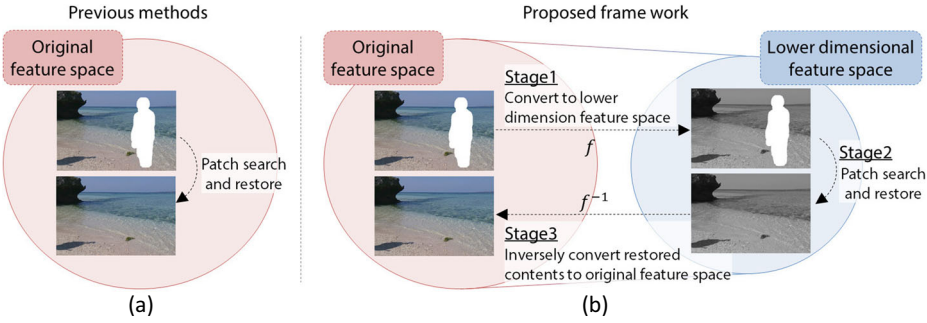

Fig. 1 Completion process of previous methods (**a**) and proposed framework (**b**). Most previous methods complete contents in the original feature space, while the proposed framework completes them in a lower dimensional feature space. Damaged contents are converted to the lower dimensional feature space in Stage 1, restored in Stage 2, and inversely converted to the original space in Stage 3

these methods take such deformations or illumination changes into account, patch retrieval requires huge computational cost.

The second one is to increase the patch availability by retrieving patches with relaxed constraints. Shiratori et al. proposed a method for restoring video in a motion field [16]. It retrieves patches on the basis of motion vectors, which makes it possible to fill in a damaged region if a motion pattern similar to that of the damaged region is contained in the reference video. Once the motion vector is restored, the missing pixel values can be obtained from the temporally neighbouring video frames. This can be done regardless of the color of the patches; i.e., the applicability of patches is extended with respect to color. However, it requires manual selection of a reference video that includes a motion pattern similar to that of the damaged region. The necessity of such intervention deteriorates the efficacy of the method in practice. In addition, when the duration of damage gets longer, small differences between the selected motion vectors and the desired ones make color propagation more difficult. We think that Shiratori et al.'s method can be considered a reasonably effective one as it restores damaged regions via a different feature space in process step (2). However, a motion vector is not always the optimal feature and in some cases another feature space is more suitable. Also, this method can be applied to video restoration only.

We propose a general framework for completion via a different feature space, which allows us to use various feature spaces. In particular, we use a lower dimensional feature space because we assume that patches in a lower dimensional feature space enhance patch availability.

2.2 Content transportation to different feature space

Studies have been made on generative approaches to content restoration, in which features are created and added to a content. Levin et al. proposed a colorization method [12] that adds color information to monochrome contents. It works under the following simple assumption: "neighboring pixels in space-time that have similar intensities should have similar colors". Therefore, color information can be estimated by solving an optimization problem formalized on the basis of this assumption and using sparsely designated color information.

Hertzmann et al. proposed a method called "image analogies". It estimates an image filter applied to a reference image and then applies it to another image to add effects similar to those of the reference to the other image [8].

Our proposed method omits some content features and completes the content using the others. It then compensates for the missing features within the completed content by using generative approaches.

3 Proposed method

We propose a novel content completion framework that consists of three stages: converting a target image to a lower dimensional feature space, restoring damaged regions in the space, and inversely converting them to the original feature space. The motivation for converting an image to a lower dimension is "to make dissimilar patches similar" by projecting to a lower dimensional feature space. Hereafter, we distinguish the words "restoration" and "completion" as follows: "restoration" is used for the second stage, restoring an image in a low dimensional feature space, while "completion" is used for all three stages including the restoration stage. In this section we first overview the framework in Section 3.1 and then in

Section 3.2 describe how to inversely convert the restored image from the lower dimensional feature space to the original feature space in Stage 3.

3.1 Proposed framework

Unlike existing methods, which restore damaged regions in an original feature space such as an RGB space (Fig. 1a) or a higher dimensional space by adding edges or motion vectors, the proposed framework uses a lower dimensional feature space. Even if there are no similar patches in the original feature space, the lower dimensional feature space in which some information is lost makes dissimilar patches become similar. The framework outline is shown in Fig. 1b. The details of each stage follow.

Stage 1. Converting input contents Input contents including damaged region I_{in} are converted from the original feature space to $I_{in'}$, which is in a lower dimensional feature space. This can be written by

$$I_{in} = f(I_{in'}) \tag{1}$$

where f is the projection function. Ideally, this conversion should excludes features that are less important for restoration or that can be compensated for by post processing. However, such features depend not only on the human vision system but on a target content. Thus, we can try making use of various lower dimensional feature space simultaneously. Here we show two examples for dimension reduction. To convert to gray scale space (one dimensional space) or RG space (two dimensional space), f can be write as following (2) and (3). Note that $I_{in'_R}$, $I_{in'_G}$, and $I_{in'_B}$ represent each RGB channels.

$$I_{in} = f(I_{in'}) = \begin{bmatrix} 0.299 & 0.587 & 0.114 \end{bmatrix} \begin{bmatrix} I_{in'_R} \\ I_{in'_G} \\ I_{in'_B} \end{bmatrix} \tag{2}$$

$$I_{in} = f(I_{in'}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_{in'_R} \\ I_{in'_G} \\ I_{in'_B} \end{bmatrix} \tag{3}$$

Stage 2. Restoration in lower dimensional space $I_{in'}$ is restored to generate I_c in the lower dimensional space.

$$I_c \longrightarrow I_{in'} \tag{4}$$

We expect that the restoration is easier in a lower dimension feature space because some patches that are not similar and cannot be used for restoration in the original feature space become similar and become available by projecting to lower dimensional feature space. Any exemplar-based restoration methods are acceptable for our framework. This is true even for video as the restoration target.

Stage 3. Inverse conversion of restored content Inverse conversion (see Section 3.2 below) is performed to obtain final output in the original feature space. Restored contents in lower feature space I_c are inversely converted to those in original feature space I_{out} as follows.

$$I_{out} = f^{-1}(I_c) \tag{5}$$

where f^{-1} is inverse projection function. This inverse conversion is necessary to compensate for the features that were omitted in Stage 1. More details for f^{-1} are explained in Section 3.2.

This three-step solution generates two advantages. First, it enhances the possibility of applying patches dissimilar to those in the original color space. Second, it enables the use of many existing methods for restoration because the feature space for retrieving the similar patches is the only extension.

Figure 2 explains how our framework works well. Here, Fig. 2a is an example of a damaged image that has no appropriate patches for restoration. For the damaged yellow button, there are no similar patches having the same structure and same color. Thus, if the original feature space is used, the blue box, which contains an orange button, is retrieved as the most similar one as in Fig. 2b. However, as shown in Fig. 2c, because yellow and pink buttons are converted to similar levels in gray scale, dissimilar patches consisting of pink buttons in RGB space become similar. This enables a region whose original color is pink to become applicable for restoration as shown in Fig. 2d.

3.2 Inverse conversion of restored content

Because Stage 1 excludes certain features, I_c (the restored result in the lower dimensional space) and I_{out} (the completed result in the original space) have a one-to-many relationship. Therefore, inverse conversion and compensation f^{-1} for missing information are required. We perform inverse transformation by using two different approaches, described in Sections 3.2.1 and 3.2.2 below.

3.2.1 Inverse conversion based on correspondence between two contents

This section describes a versatile approach utilizing data gotten from content pairs, i.e., the original content I_{in} and the converted content $I_{in'}$. Note that getting such data is easy because they exist in non-damaged areas in content pairs. By using such pairs, Stage 3 infers the inverse conversion from the non-damaged areas of content pairs. Because this approach can be used regardless of the converted lower dimensional feature space, it becomes a versatile approach.

We use Image Analogies [8] to implement the idea described above because it works well regardless of the number of samples and distribution of the data. A more detailed process, which has four steps, is as follows (See Fig. 3). First, multi-scale representations of I_{in} and $I_{in'}$, before and after Stage 1, and the restored result in lower dimensional space

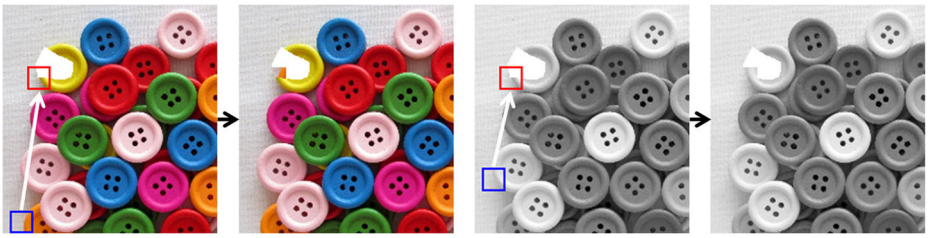

Fig. 2 An example of how our framework contributes to a completed result, where (**a**) shows a damaged original image (damaged region is masked in white). In the original RGB space, an inappropriate similar patch (shown as a blue box) is retrieved for a damaged patch (shown as a red box), which results in completion failure as shown in (**b**). However, as shown in (**c**), yellow and pink buttons are converted to similar levels in gray scale, which enables a region whose original color is pink to become applicable for restoration as in (**d**)

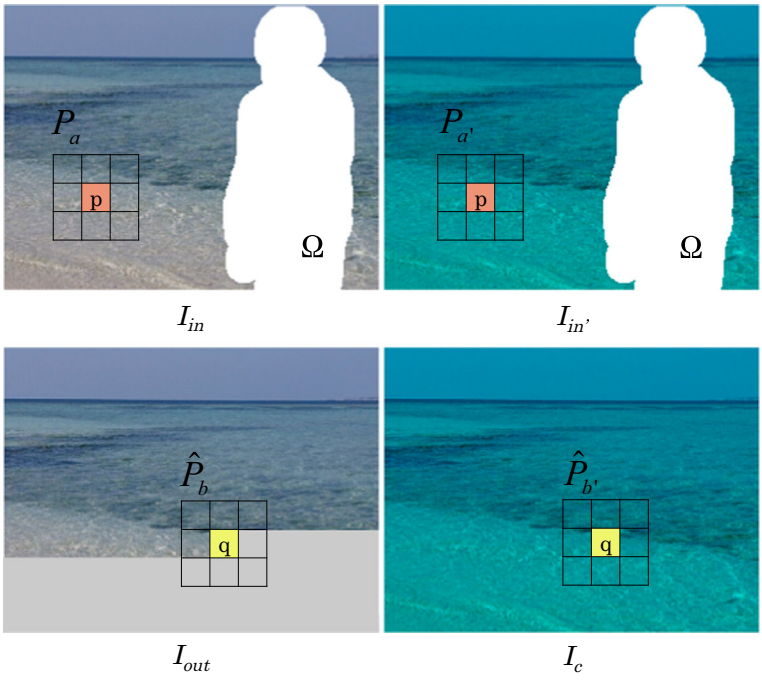

Fig. 3 Inverse conversion from the lower dimensional feature space to the original space via Versatile. Data vectors based on $P_a(p)$ and $P_{a'}(p)$, as well as patches centered at every pixel p in I_{in} and $I_{in'}$ are stocked as a database for conversion. Pixel q in restored content I_c is converted using a similar data vector to $V(q)$, a data vector based on $\widehat{P}_{b'}(q)$ and $\widehat{P}_b(q)$ and patches centered at q in I_c and I_{out}

I_c are constructed. Data vector $V(p)$ including information of I_{in} and $I_{in'}$ is then stocked for every non-damaged pixel p. $V(p)$ consists of $P_a(p)$ and $P_{a'}(p)$, which correspond to patches centered at p in I_{in} and $I_{in'}$. After that, I_c is inversely converted to I_{out} on a pixel-by-pixel basis. To convert pixel q in I_c, data vector $V(q)$ including information of $\widehat{P}_b(q)$ and $\widehat{P}_{b'}(q)$ is calculated, where $\widehat{P}_b(q)$ and $\widehat{P}_{b'}(q)$ are patches centered at q in I_{out} and I_c. Finally, similar data vector $V(p)$ of $V(q)$ is retrieved from the database and q is updated by p.

For video content, this algorithm also works well by processing frame-by-frame or by extending $V(p)$ consisting of spatially neighboring pixels to spatio-temporal neighboring ones.

3.2.2 Dedicated inverse conversion

This section describes a dedicated method for each feature space. It is not a general-use method but a specialized one for feature spaces that is expected to enable better transformation.

One example method of this type is colorization [12], which is effective in generating lost color. It can be used when gray scale feature space is used for restoration. Let us examine colorization-based inverse conversion in more detail. Colorization needs color information seeds within a gray image to be colorized and many previous studies, including [12], set

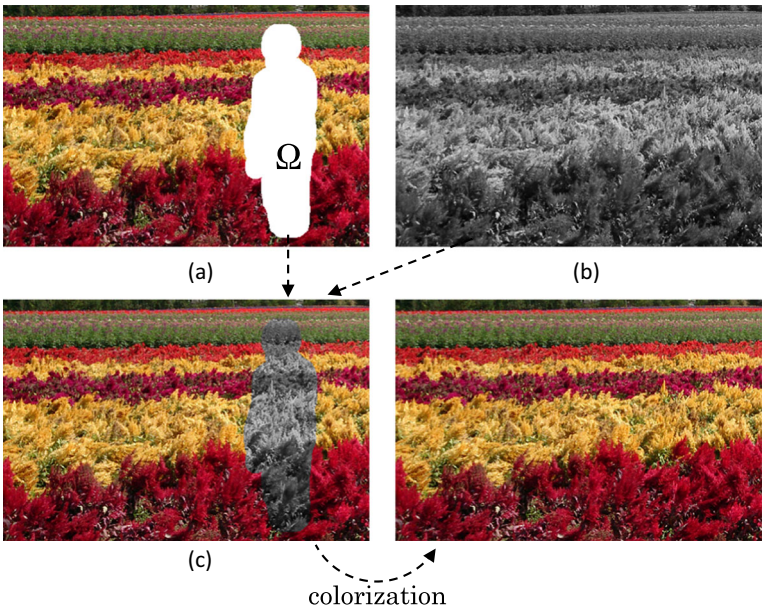

Fig. 4 Initialization for colorization process. Initial content (**c**) is generated by seeding color information from non-damaged regions in I_{in} shown in (**a**), to restored content I_c shown in (**b**)

such seeds manually. However, in our case seeding can be automated because color information of non-damaged regions exists in I_{in}. As shown in Fig. 4, initial color values for colorization are set by using the color values of the original image (Fig. 4a) as follows, where Ω represents the damaged region in the image:

$$I_c(x) = \begin{cases} I_c(x) & (x \in \Omega) \\ I_{in}(x) & (otherwise) \end{cases} \tag{6}$$

Consequently, in our implementation we consider there are patches $P(p)$ centered at every damaged pixel p. Color information of p is estimated by solving an optimization problem so that p and its neighboring pixels in $P(p)$ keep luminance consistency. This process is also effective for setting $P(p)$ as a 3D patch including spatio-temporal neighbor pixels.

Another example method of this type is super resolution-based inverse conversion. This method is effective when a low resolution space is used as the restoration space. There are many possible implementations with the existing algorithm.

4 Experiment

This section demonstrates how the proposed method works, i.e., how it improves completion while maintaining the advantages of previous restoration methods. To simplify the discussion, this section only focuses on one simple implementation, i.e., applying gray scale conversion in Stage 1, performing restoration in gray space in Stage 2, and colorizing in Stage 3. We apply this implementation with the expectation that unsuitable patches that have an appropriate structure but inappropriate color can be used for restoration in gray scale feature space.

In Section 4.1 we describe the restoration methods applied in Stage 2 and in Section 4.2 we describe in detail how our method works and show completed results with calculation times and an objective evaluation.

4.1 Restoration methods

For the content restoration in Stage 2 we used two methods for image [3, 5], and one method for video [13]. All of them restore a damaged region on the basis of the similar patches retrieved. However, they use the retrieved patches in different ways and thus derive different advantages. In this subsection, we introduce these three techniques in more detail.

Criminisi et al.'s method [3] is based on the idea of copying and pasting of small patches from a source area into the damaged region Ω. These patches are useful as they provide a practical way of encoding local texture and structure. The method does not guarantee global coherence, but it includes a way to propagate both linear structure and texture into the hole region from patches with highest priority. The priority computation is biased toward patches that are (i) on the continuation of strong edges and (ii) are surrounded by high-confidence pixels. Given a patch Φ_p centered at point p for multiple p included in the contour of the damaged region, they define priority $P(p)$ as below.

$$P(p) = C(p)D(p) \tag{7}$$

Here, $C(p)$ and $D(p)$ correspond to (i) and (ii), respectively. They are defined as follows:

$$C(p) = \frac{\sum_{q \cap \bar{\Omega}} C(q)}{|\Phi_p|}, D(p) = \frac{|\Delta I_p^{\perp} \cdot n(p)|}{\alpha} \tag{8}$$

where $n(p)$ is a vector orthogonal to the contour of the damaged region, and $I_p^{\perp}$ is computed as the maximum value of the image gradient in $\Phi_p \cap \bar{\Omega}$. α is a normalized factor, to be set as 255 for a typical image.

Efros et al.'s method [5] efficiently restores holes included in periodic texture content. With this method, the damaged region to be filled is synthesized one pixel at a time. To synthesize a pixel p, the algorithm first finds patches $\mathbf{w}(p)$ from the neighboring area in the sample image that are similar to $\mathbf{P}(p)$, i.e., patches including p. It then chooses one neighborhood patch $\mathbf{w}_{best}(p)$ from $\mathbf{w}(p)$ to minimize a difference between $P(p)$ and $\mathbf{w}(p)$ as follows.

$$\mathbf{w}_{best}(p) = argmin\ distance(P(p), \mathbf{w}(p)) \tag{9}$$

Positions of p within $\mathbf{w}_{best}(p)$ are represented as x_p. Finally, p is newly synthesized using x_p as a basis.

Newson et al.'s method [13] is effective for video content. This method restores the damaged region on a pixel-by-pixel basis. First, several patches $\mathbf{w}(p)$ including damaged pixel p are set as target patches. Positions of p within these patches are represented as $\mathbf{x}$. Similar patches $\widehat{w}$ for each $\mathbf{w}$ are then retrieved. Finally, p is updated on the basis of the weighted mean value of q with the following formula:

$$u_p = \frac{\sum_{q \cap \bar{\Omega}} s_q u_q}{\sum_{q \cap \bar{\Omega}} s_q}, q = \{x \in \widehat{w}\} \tag{10}$$

where Ω is the damaged region, u_p and u_q represented the RGB values of p and q, and s_q is a weighted value for q.

In this paper, using the previous algorithms as a basis, we represent the proposed method as *Prop.Method(Feature)*. For example, we represent the proposed method whose

lower feature space is RG feature space and which is based on Criminisi et al.'s work as *Prop.Criminisi(RG)*.

4.2 Completion and evaluation results

In this section we demonstrate completed image results in Section 4.2.1 and video results in Section 4.2.2. In Section 4.2.3 we show objective evaluations of the methods used in the study in terms of calculation time and similarities.

4.2.1 Image completion result

Here we show the completion results obtained with the proposed methods and compare them to results obtained with their restoration methods, Criminisi et al. [3] and Efros et al. [5]. Two completed target images including a complex structure and color changes were used for this comparison. The first one, shown in Fig. 5I, has a rather large damaged region. Therefore, we consider that Criminisi et al.'s method is suitable for restoring it. The other one, shown in Fig. 5II, has a smaller damaged region but a unique cyclic structure. We considered that for this kind of cyclic structure, Efros et al.'s method would be better. Note that both target images include a complex structure and would be difficult to restore with methods using spatial consistency.

Additional experimental settings are as follows. The damaged region is manually set (Fig. 5a masked in white), the image resolutions are (I) 210×223 and (II) 200×150 pixels, the ratios of damaged pixels in each image are (I) 1.77 % and (II) 2.91 % and the patch sizes we used are (I) 21×21 pixels and (II)13×13 pixels. To perform the experiments we used a desktop PC of Intel Core i7 3.40GHz CPU, 32GB memory, and a Matlab R2014. The results obtained with the proposed methods *Prop.Crimininisi(gray)* and *Prop.Efros(gray)* are shown in Fig. 5d and e. Those obtained with the previous methods (Criminisi et al.'s and Efros et al.'s) are shown in Fig. 5b and c.

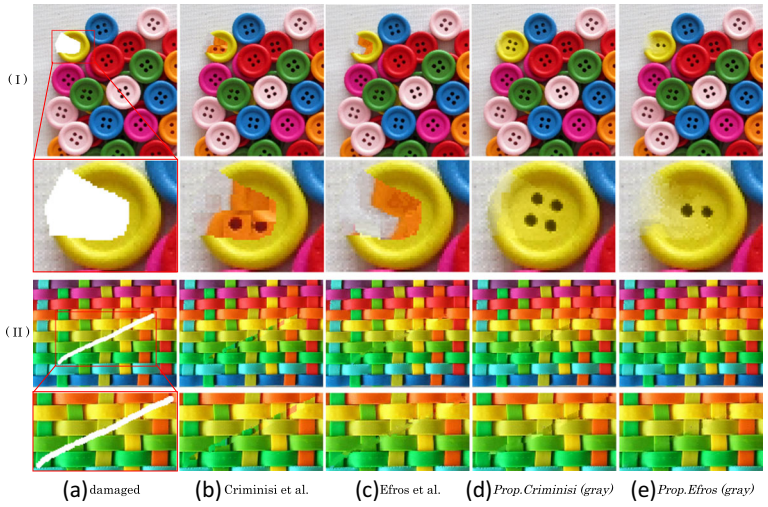

Fig. 5 Input and result of image completion experiment; (**a**) Input image with damaged region masked in white. The close-up area is shown as a red frame. (**b**), (**c**) Results obtained by Criminisi et al. [3] and Efros et al. [5]. (**d**), (**e**) Results obtained with the proposed method: *Prop.Criminisi(gray)* and *Prop.Efros(gray)*

In Fig. 5a the top row (I) shows the completed results obtained with the methods used. As can readily be seen, *Prop.Criminisi(gray)* showed the most efficient completion. Comparing the *Prop.Criminisi(gray)* and *Prop.Efros(gray)* results makes it clear that the proposed framework well maintains the advantages of a base restoration method. Because this completion target (Fig. 5a) includes a rather large damaged region, it is intrinsically suitable for Criminisi et al.'s method.

From Fig. 5II it is clear that Efros et al. in (c) shows better performance than Criminisi et al. in (b), indicating Efros et al.'s method is advantageous for dealing with periodic structured content. Some unnatural shadows are observed, however, especially on the yellow-green warp at the center and on the yellow warp next to that. In contrast, *Prop.Efros(gray)* did not show any such defects (e). These results well verify that the proposed framework retains the advantages of a base restoration method and improves completion quality.

Although for explanatory purposes we used unnatural images for the completion target, rather primitive restoration methods, and gray scale space for restoration, we will show more comprehensive completion results in Section 5.

4.2.2 Video completion results

For obtaining video completion results, we implemented *Prop.Newson(gray)* as the proposed method, using Newson et al.'s [13] algorithm as a basis, via gray scale feature space. Figure 6 shows a comparison between *Prop.Newson(gray)* and Newson et al.'s method. The target sequence has 104 frames with 960 × 540 pixel resolution. The damaged region is automatically designated and its average percentage is 6.5 % of the original video.

Completed results obtained with Newson et al. are shown in the second row of Fig. 6. The result for *Prop.Newson(gray)* is shown as a restored sequence in gray scale feature space in the third row. The final result in the original color space, inversely converted by colorization, is shown in the bottom row. With Newson et al's method, an easily distinguished red colored area appeared in the bottom area in the enlarged images. For the same area, *Prop.Newson(gray)* achieved completing with natural water color.

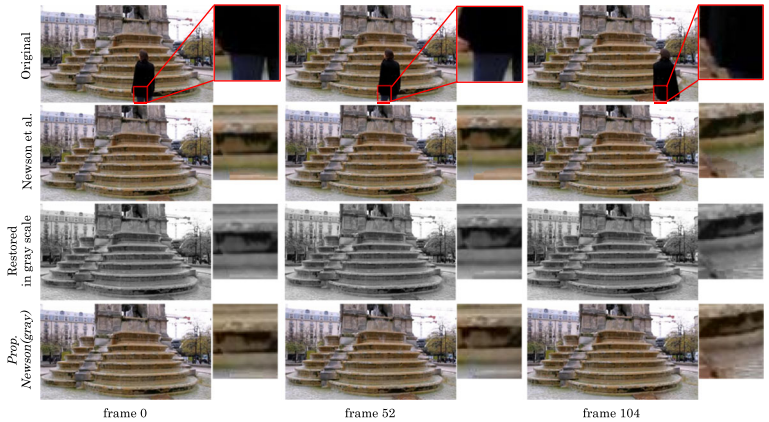

Fig. 6 Original frames and video completion results. Original frames including an unwanted area are shown in the top row and completed results obtained with Newson et al. are shown in the second row. In our implementation, we first obtained the restored results in low-dimensional gray space (third row) and then generated the final results by colorizing them as shown in the bottom row

Table 1 Evaluation by SSIM with default parameters of [18]

	Size	Criminisi et al.	Efros et al.	*Prop.Criminisi(gray)*	*Prop.Efros(gray)*
(I)	original	0.9824	0.9826	0.9894	0.9828
	close-up	0.6451	0.6463	0.7857	0.6500
(II)	original	0.9780	0.9834	0.9828	0.9937
	close-up	0.9368	0.9528	0.9517	0.9817

The highest scores are underlined

4.2.3 Objective evaluation

Objective evaluations were made among the methods in terms of similarities and computational cost. For evaluation purposes, we calculated SSIM (Structure SIMilarity) [18] and PSNR[2](Peak Signal-to-Noise Ratio) for the (I) (II) results in Fig. 5. SSIM is a metric for using structure information to calculate image similarity. It is a decimal value between -1 and 1, with 1 being the highest score. The comparative results for the methods used are shown in Tables 1 and 2. We calculated these values for original size images and also for the close-up view in Fig. 5. As the tables show, in terms of (I) *Prop.Criminisi(gray)* recorded the highest value and *Prop.Efros(gray)* recorded the next highest score for both SSIM and PSNR. With respect to (II), *Prop.Efros(gray)* showed the highest value and *Prop.Criminisi(gray)* scored the second highest for both SSIM and PSNR.

Also, although our implementation, so far, does not focus on reduction of calculation cost, we briefly examined elapsed time for processing. Table 3 shows a comparison of elapsed time between Criminisi et al.'s method and *Prop.Criminisi(gray)*, and Table 4 shows the same between Efros et al.'s method and *Prop.Efros(gray)*. There was no significant difference between the elapsed time of Criminisi et al.'s method and *Prop.Criminisi(gray)*, despite the fact that *Prop.Criminisi(gray)* requires an additional process, i.e., colorization. The comparison between Efros et al's method and *Prop.Efros(gray)* (Table 4) shows that the calculation time was much less for the latter. The calculation cost of the former is high because of the pixel-by-pixel restoration it performs. The calculation time for *Prop.Efros(gray)* is lower because the completion was done in a lower dimensional space. Of course the calculation cost will change depending on hole size or the initialization required for colorization, but this evaluation confirmed that performing inverse conversion does not significantly affect the total calculation time for smaller images such as those shown in Fig. 5. To further elaborate on this point, Section 6.2 describes how we analyzed computational cost in more detail.

5 Results in various settings

In this section, we show that our framework is also effective with current state-of-the-art algorithms and other feature spaces. The restoration method and feature space we used are shown in Table 5. Their details are as follows.

[2]Target PSNR for general purpose lossy image compression ranges from 30 dB to 50 dB, where the higher is the better.

Table 2 Evaluation by PSNR[1] [dB]

	Size	Criminisi et al.	Efros et al.	*Prop.Criminisi(gray)*	*Prop.Efros(gray)*
(I)	original	28.75	27.36	32.46	29.38
	close-up	16.94	15.55	20.65	17.57
(II)	original	29.72	30.01	33.82	38.55
	close-up	25.30	27.62	29.41	34.14

The highest scores are underlined

5.1 Restoration method

First, we introduce He et al.'s and Huang et al.'s algorithm [7, 9] for Stage2, the image restoration part.

He et al.'s method [9] works well for filling in missing regions through patch offset statistics. If similar patches in the image are matched and their relative positions obtained, the statistics of these offset areas are sparsely distributed. With these offsets the missing regions are filled by combining a stack of shifted images via photomontage, a method for image composite by using optimization.

Huang et al.'s method [7] is also a current state-of-the-art method and works especially well with images including complex structures. It first estimates perspective and regularity in the source image and roughly segments the known region into planes, then discovers translational regularity within these planes. The information is then converted into soft constraints for the low-level restoration algorithm by defining prior probabilities for patch offsets and transformations.

5.2 Feature space

We use three types of feature space: RG, GB, and gray scale space. An RG feature space is a color space that has red and green channels only. A GB space has green and blue channels. RG and GB spaces are represented as two dimensions while original the RGB space has three dimensions. We expect that the availability of patches will be increased by using these feature spaces because of the decrease in dimensions. We also expect that visually important features will remain in RG and GB spaces more than in gray scale, which is represented as one dimension. Many feature spaces are represented as two dimensional spaces, but the green channel is well known as visually important information. It is for this reason that we use RG and GB spaces, which include green information. As an inverse conversion process

Table 3 Elapsed time comparison between Criminisi et al.'s method [3] and *Prop.Criminisi(gray)* [sec]

	Criminisi et al.	*Prop.Criminisi(gray)*		
		Completion	Colorization	Total
(I)	1.64	0.87	1.11	1.98
(II)	2.19	1.31	0.40	1.71

For more detail, please see Section 6.2

Table 4 Elapsed time comparison between Efros et al.'s method [5] and *Prop.Efros(gray)* [sec]

	Efros et al.	*Prop.Efros(gray)*		
		Completion	Colorization	Total
(I)	297.00	147.19	0.48	147.67
(II)	77.47	46.97	0.28	47.25

For more detail, please see Section 6.2

for gray scale space, we used colorization [12] in the same way as mentioned in Section 4.2. For RG space and GB space the versatile method described in Section 3.2.1 was used.

5.3 Results

Completed results for the natural scenes we used are shown in Fig. 7. Original images with damaged regions (masked in red) are shown in column (a), while (c) shows the completed results obtained with the proposed method. We show the most effective results that were obtained with various feature spaces. The restored results obtained with the base restoration method used to get the results in (c) are shown in (b). The proposed completion method shows better performance than the other methods because of its utilizing a lower dimensional feature space for restoration.

6 Limitation and future work

6.1 Feature space selection

So far, we have showed the results obtained in using a specific lower dimensional feature space for restoration without explanation. Figure 8 shows completion results obtained using the same restoration method but a different feature space; original images with the damaged region masked in red are shown in column (a). Effective and ineffective results obtained by using different feature spaces are shown in (b) and (c). Note that the only difference between them is the feature space used for restoration; the same restoration method was used for both.

Table 5 Variations of restoration methods and feature spaces

Restoration methods	Feature spaces		
	Gray scale	RG	GB
He et al [9]	*Prop.He(gray)*	*Prop.He(RG)*	*Prop.He(GB)*
Huang et al [7]	*Prop.Huang(gray)*	*Prop.Huang(RG)*	*Prop.Huang(GB)*

Note that all of the patterns with "*Prop.*" were obtained with our proposed methods

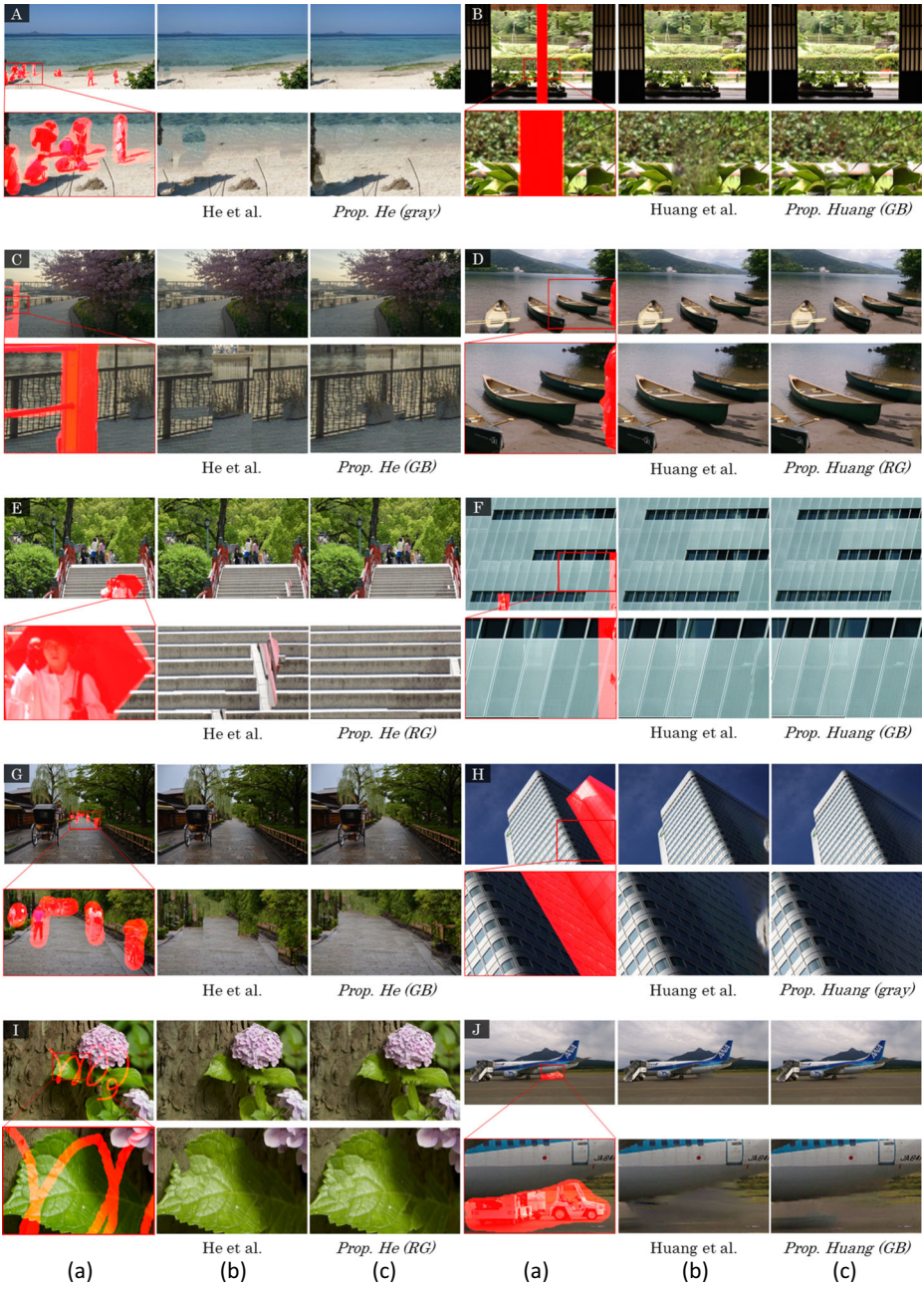

Fig. 7 Completed results obtained with current state-of-the-art restoration methods and various feature spaces (gray, RG, and GB). (**a**) Target images with damaged regions (masked in red). (**b**) Results obtained with the original restoration method without dimension reduction. (**c**) Results obtained with the proposed method with dimension reduction. Note that all of the image results annotated "*Prop.*" were obtained with our proposed methods

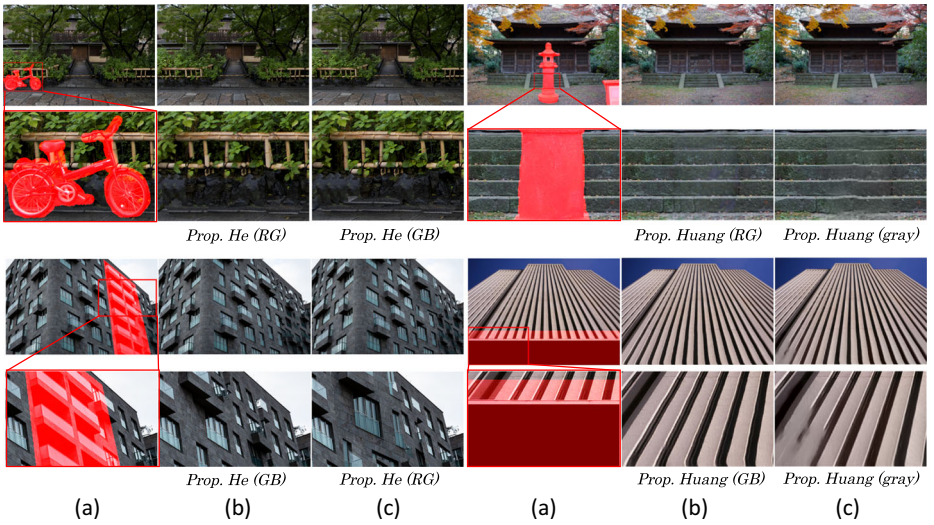

Fig. 8 Comparison between completed results obtained using different feature spaces for restoration. Note that the same restoration method was used to obtain the two results. (**a**) Target images with damaged region masked in red. (**b**), (**c**) Completed results obtained with effective/ineffective feature space

As the results show, completion performance depends on the feature space used for restoration. It was already mentioned in Section 4 that different restoration methods produce different completion results even if the same feature space is used for restoration. Thus, completion performance is affected not only by the feature space used for restoration but by the restoration method. Currently, however, we have not established any criteria for selecting an appropriate feature space and restoration method before observing the completion results. One possible solution is to show the completion results obtained using various setups, i.e., combinations of different restoration methods and feature spaces, and to have users perform the task of selecting from among the completion results.

However, this is likely to make things difficult for the users because of the large number of possible combinations. Aiming to provide the best inpainting result automatically, we have already started developing an automatic ranking method for inpainted results reflecting subjective preference of them [10]. This enables to provide the best result from a set of images inpainted by various methods and feature spaces. Our next step will be to reveal the optimal combination of method and feature only by a completion target image.

6.2 Calculation cost

In Section 4.2.3, we show an example of elapsed time. However, introducing our proposed framework produced no significant changes in elapsed time (in that setting, we used the dedicated method version of inverse conversion for Stage 3, as given in Section 3.2.2). Although

Table 6 Elapsed time comparison between He et al.'s method [9] and *Prop.He(gray)* with two inverse conversion method explained in Sections 3.2.1 and 3.2.2 [sec]

He et al.	*Prop.He(gray)*				
	Completion	inverse conversion via Section 3.2.1		inverse conversion via Section 3.2.2	
		inverse conversion	total	inverse conversion	total
19.1	14.4	1513.6	1528.0	31.6	46.0

we do not focus on the issue of calculation cost in this paper, we think the proposed framework has potential for accelerating the processing time of completion. Therefore, we analyze the computational cost to further elaborate on this point.

In general patch-based completion methods like that in [3], the patch retrieval process occupies most of the total calculation time. Its calculation order is $\mathcal{O}(NM^Z)$, where N and M are respectively the number of missing pixels and the number of total pixels in the image. The parameter Z is a feature vector dimension for retrieving. This dimension affects the total calculation order with the power of Z, and thus the proposed method, which uses a lower dimensional space for restoration, may reduce the total calculation cost.

At the same time, the method needs to perform additional processing subsequent to restoration, i.e., inverse conversion of feature space from a lower dimension to the original one, as shown in Section 3.2. For this we introduced two approaches, a generalized approach in Section 3.2.1, and dedicated approach in Section 3.2.2. The former requires considerable computational cost. Table 6 shows an example of elapsed time for (b) and (c) in Fig. 7 A with the inverse conversion method presented in Sections 3.2.1 and 3.2.2. The completion time for *Prop.He(gray)* is lower than that for the original method because the restoration was done in a lower dimensional space. However, because of the larger resolution of Fig. 7 A (1280×720)[pixels], the inverse conversion methods in Sections 3.2.1 and 3.2.2 (particularly the former) entail quite high calculation cost.

We think that these inverse conversion methods can be made much faster by implementing parallel computation (e.g. with GPU), because these algorithms enable parallel computing to be performed relatively easily. The time required for users to mask unwanted regions by users should also be taken into account. Therefore, a subject for future work will be to consider how to reduce the time required for inverse conversion and masking.

7 Conclusion

In this paper, we introduced a new framework for image/video completion. Our framework involves three stages: (1) converting input content to a lower dimensional feature space, (2) restoring the content in the converted lower dimensional space, and (3) inversely converting the restored content from the lower dimensional space to the original feature space. We consider the framework to be an effective approach, first because it enhances the possibility

of applying patches dissimilar to those in the original color space, and second because it makes it possible to use a variety of restoration methods and feature spaces. Experiment results have verified its effectiveness.

Currently we have not established any criteria for selecting an appropriate feature space and restoration method before observing completion results. Thus, subjects for future work will include developing a method that will enable the most appropriate results to be selected automatically, with which we are currently working, and developing criteria that will allow restoration methods and features to be selected before the completion process is carried out.

References

1. Bertalmío M, Sapiro G, Caselles V, Ballester C (2000) Image inpainting. In: Proceedings of the ACM SIGGRAPH, pp 417–424
2. Bertalmio M, Vese L, Sapiro G, Osher S (2003) Simultaneous structure and texture image inpainting. IEEE Trans Image Process 12(8):882–889
3. Criminisi A, Perez P, Toyama K (2004) Region filling and object removal by exemplar-based inpainting. IEEE Trans Image Process 13(9):1200–1212
4. Darabi S, Shechtman E, Barnes C, Goldman DB, Pradeep S. (2012) Image Melding: Combining inconsistent images using patch-based synthesis. ACM Trans Graph (TOG) (Proceedings of SIGGRAPH 2012) 31(4):82:1–82:10
5. Efros AA, Leung TK (1999) Texture synthesis by non-parametric sampling. In: Proceedings of International Conference on Computer Vision (ICCV), vol 2, pp 1033–1038
6. Herling J, Broll W (2014) High-quality real-time video inpainting with pixmix. IEEE Trans Vis Comput Graph 20(6):866–879
7. Huang J-B, Kang SB, Ahuja N, Kopf J (2014) Image completion using planar structure guidance. ACM Trans Graph (Proceedings of SIGGRAPH 2014) 33(4):129:1–129:10
8. Hertzmann A, Jacobs CE, Oliver N, Curless B, Salesin DH (2001) Image analogies. In: Proceedings of the ACM SIGGRAPH, pp 327–340
9. He K, Sun J (2014) Image completion approaches using the statistics of similar patches. IEEE Trans Pattern Anal Mach Intell 36(12):2423–2435
10. Isogawa M, Mikami D, Takahashi K, Kojima A (2015) Toward enhancing robustness of dr system Ranking model for background inpainting. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp 178–179
11. Kawai N, Machikita K, Sato T, Yokoya N (2011) Video completion for generating omnidirectional video without invisible areas. Inf Media Technol 6(1):158–171
12. Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. In: Proceedings of the ACM SIGGRAPH, pp 689–694
13. Newson A, Almansa A, Fradet M, Gousseau Y, Pérez P (2014) Video inpainting of complex scenes. SIAM J Imaging Sci 7(4):1993–2019
14. Sun J, Lu Y, Jia J, Shum H-Y (2005) Image completion with structure propagation. ACM Trans Graph (Proceedings of SIGGRAPH 2005) 24(3):861–868
15. Shih TK, Tang NC, Hwang J-N (2009) Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. IEEE Trans Circ Syst Video Technol 19(3):347–360

16. Shiratori T, Matsushita Y, Tang X, transfer SingBingKang. (2006) Video completion by motion field. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp 411–418
17. Wexler Y, Shechtman E, Irani M (2007) Space-time completion of video. IEEE Trans Pattern Anal Mach Intell 29(3):463–476
18. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

Mariko Isogawa received her B.S. and M.S. degrees from Osaka University, Japan, in 2011 and 2013, respectively. She has been working for Nippon Telegraph and Telephone Corporation from 2013. Her research interests include multimedia content handling.

Dan Mikami received his B.E and M.E degree from Keio University, Kanagawa, Japan in 2000 and 2002, respectively. He has been working for Nippon Telegraph and Telephone Corporation from 2002. He received his Ph.D. from Tsukuba University in 2012. His current research activities are mainly focused on multimedia content handling. He was awarded the Meeting on Image Recognition and Understanding 2009 Excellent Paper Award 2009, the IEICE Best Papar Award 2010, the IEICE KIYASU-Zen'iti Award 2010, and the IPSJ SIG-CDS Excellent Paper Award 2013. He is a member of IEICE, IPSJ, and IEEE.

Kosuke Takahashi received the B.Sc. degree in engineering and the M.Sc. in informatics from Kyoto University, Kyoto, Japan, in 2010 and 2012, respectively. He is currently a researcher at NTT Media Intelligence Laboratories. His research interests include computer vision. He received "Best Open Source Code" award Second Prize in CVPR 2012.

Akira Kojima received the B.E. and M.E. degrees in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1988 and 1990, respectively. Since joining NTT in 1990, he has been engaged in research and development on video database, digital library, multimedia information retrieval, video surveillance, and high-reality visual communication. He is currently a Senior Research Engineer, Supervisor, Visual Media Project, NTT Media Intelligence Laboratories. Mr. Kojima is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), the Institute of Image Electronics Engineers of Japan (IIEEJ), and Association for Computing Machinery (ACM).