Usability-Aware Image Inpainting

Mariko Isogawa

March 2019

Usability-Aware Image Inpainting

A dissertation submitted to
THE GRADUATE SCHOOL OF ENGINEERING SCIENCE
OSAKA UNIVERSITY
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN ENGINEERING

by
Mariko Isogawa

March 2019

Abstract

Missed or unwanted information is everywhere in visual media. Images or videos are often captured with unwanted pixels, including film grain, sensor noise, or objects in front of the filming target. Moreover, in many cases, depth images and point clouds routinely have missing holes.

Inpainting has been given an increasing amount of attention due to its ability to remove and fill in such regions. In terms of the general workflow, users indicate unwanted regions, which are to be removed, by masking them. Many methods have been proposed after the first inpainting framework was introduced in 2000 by Bertalmio et al. [1]. In recent years, inpainting has become an indispensable technique for enhancing photographs or movies. However, for many users, it is still difficult to use the technique effectively.

One reason for this is that existing inpainting methods require well-tuned parameters or masked regions. Thus, the users are required to iteratively tune parameters or mask them, employing inpainted results until desired ones are obtained by trial and error. However, this is a very time-consuming task. It is also possible that certain difficult images are not easily inpainted or no existing method can inpaint the images effectively. Most existing methods fail when similar regions do not exist in undamaged regions. General users cannot find a solution for their tasks with such method-matter failure.

To overcome these difficulties, this study explores new approaches for inpainting, thereby providing a number of contributions in this field of research. First, a method is proposed to pair-wisely estimate the better inpainted result, aiming at selecting the best one from a number of results with different parameters. Although it is not possible to estimate the inpainted quality before inpainting is performed, our method is designed to delete the parameter-tuning processes often utilized by users.

Second, to obtain training data used in pair-wise preference estimation, simulated "failed" inpainted images are used. This contributes to achieving the preferential estimation without any manual annotation. The other contributions is a way to generate degraded inpainted results.

Thirdly, this study proposes a way to automatically optimize the masked region so that the final inpainted image becomes a more "perceptually-natural" ones. This approach contributes to reducing the user's iteratively-mask setting processes.

Finally, this study addresses the general problem regarding inpainting, which most existing methods fail to address when similar regions to be used for restoration do not exist in images or database. To solve this problem, lower-dimensional space is used for the restoration process and to compensate the lost information after restoration. This new framework contributes to improving the applicability of existing restoration algorithms.

Preface

All publications that have resulted from the studies presented in this thesis are listed as follows.

Image quality assessment for inpainted images via learning to rank (Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Hideaki Kimata, *Multimedia Tools and Applications*, 2018)
This work is discussed in Chapter 2. I presented this work at related international conferences of IEEE ISMAR 2015 [2] and ICIP 2016 [3]. I wrote the manuscript, produced the supplementary media, and submitted this work, under the supervision of Dr. Mikami.

Which is the better inpainted image? Training data generation without any manual operations (Mariko Isogawa, Dan Mikami, Daisuke Iwai, Kosuke Takahashi, Hideaki Kimata, and Kosuke Sato, *International Journal of Computer Vision (IJCV)*, 2018)
This work is shown in Chapter 3. This work is also presented at selected spotlight session in BMVC 2017 [4]. I wrote the manuscript, produced the supplementary media, and submitted this work, under the supervision of Dr. Mikami, Dr. Iwai, and Dr. Sato.

Mask Optimization for Image Inpainting (Mariko Isogawa, Dan Mikami, Daisuke Iwai, Hideaki Kimata, and Kosuke Sato, *IEEE Access*, 2018)
I describe this work in Chapter 4. I also presented this work in the 21st Meeting on Image Recognition and Understanding (MIRU) in 2018 and was awarded a MIRU Interactive Session Award. I wrote the manuscript, produced the supplementary media, and submitted this work, under the supervision of Dr. Mikami, Dr. Iwai, and Dr. Sato.

Image and video completion via feature reduction and compensation (Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima, *Multimedia Tools and Applications*, 2017)
This work is introduced in Chapter 5. I presented this work in the symposium of IEICE Technical Committee on Media Experience and Virtual Environment (MVE) and was awarded Best Presentation Award (MVE Award) in 2014. This work was also presented at IEEE ISMAR 2015 [5]. I wrote the manuscript, produced the supplementary media, and submitted this work, under the supervision of Dr. Mikami.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer graphics (CG) or computer vision (CV) based image processing techniques have been researched to enhance images that we seek. Such techniques have been introduced in various applications (e.g., color enhancement, texture transfer, creating images that did not originally exist, or excluding portions of those images that exist).

Image inpainting, originally proposed by Bertalmio et al. [1] is known as one of the most remarkable techniques among them. Images sometimes include unwanted regions, such as a person walking in front of a filming target or a trash can on a beautiful beach. Inpainting has been researched to automatically removes and restores such unwanted regions in productively, so that they are as "perceptually-natural". The technique has been used in various applications, such as object removal, damaged region restoration, and missing value completion.

The general inpainting workflow is shown in Fig. 1.1. It consists of three steps: (1) users mask unwanted regions in the input image, (2) users set the parameters, and (3) users execute the inpainting process. These three steps are often repeated until the users generate their desired images. To date, numerous effective inpainting algorithms have been proposed [1, 6, 7, 8, 9, 10, 11, 12, 13, 14]. These previous studies primarily focused only on the third step, and the contributions of most of them are divided into two as shown in Tab. 1.1; robustness and calculation cost. However, inpainting still requires its users possess considerable expertise because of the following difficulties:

- Inpainting results vary largely depending on the method used and the parameters set. Thus, users are required to iteratively repeat parameter tuning and review their changes until the desired results are obtained.
- Inpainted results vary largely depending on the shape of the masked region that are indicative of the region to be restored. In a typical scenario, users have to iteratively modify a masked region until the desired results are obtained, which is often time consuming.
- Even if parameters or masked regions are well-tuned, inpainting often fails due to the limitations of the inpainting algorithm used. Hence, to avoid such failures, users are required to select appropriate data for restoration and/or select robust inpainting methods, which significantly decrease usability.

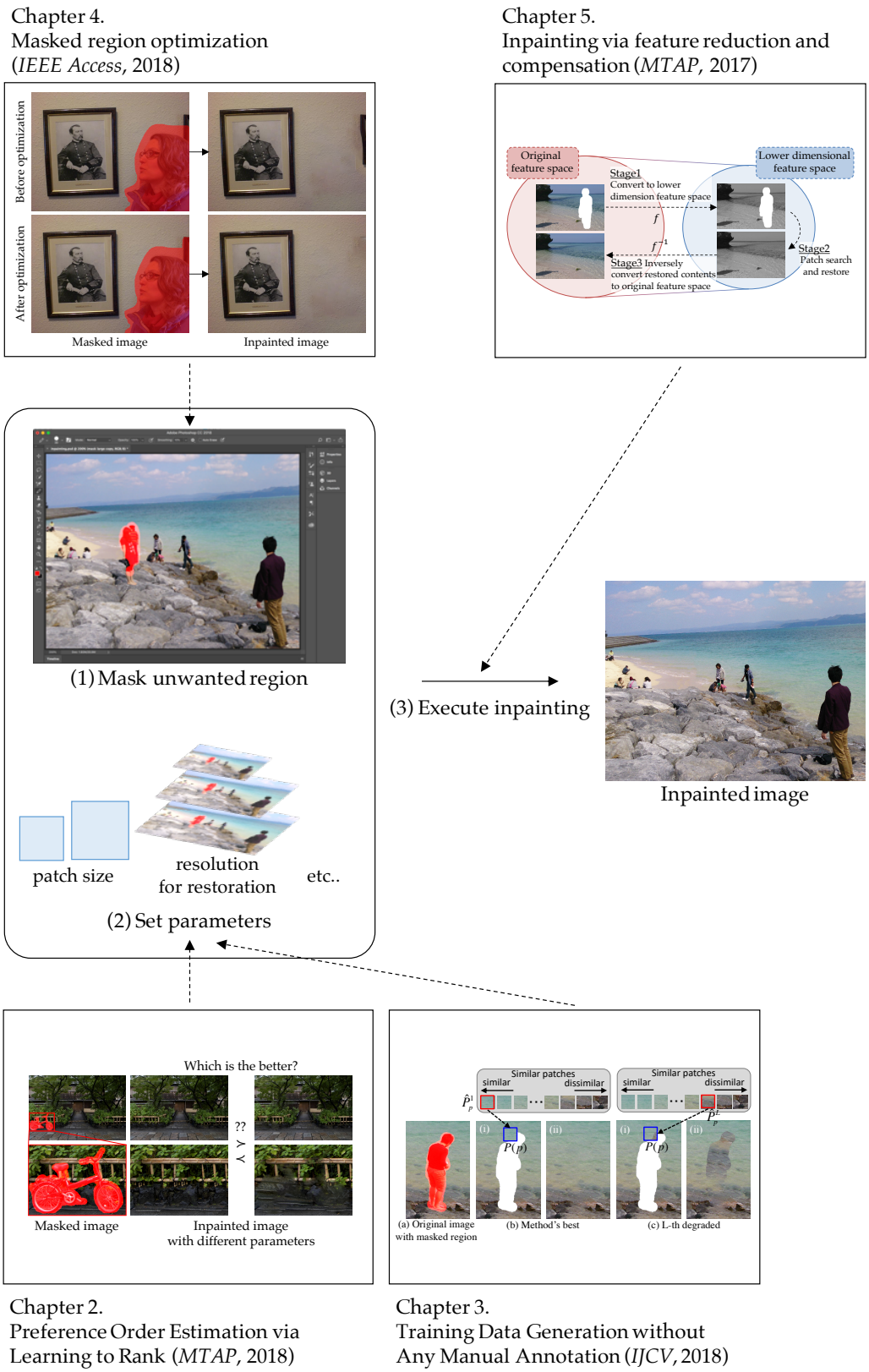


Figure 1.1: General inpainting workflow consists of three steps: (1) users indicate unwanted region by masking them, (2) tune parameters for restoration, and (3) execute inpainting. Chapter 2 to 5 of this thesis solve the problems occurred in these three steps.

Table 1.1: Main contributions of existing and our methods.

		Cost	
	Robustness	Calculation Cost	User Manipulation Cost
Bertalmio et al. [1]	✓		
A. Telea [7]		✓	
Criminisi et al. [6]	✓		
Bernes et al. [8]		✓	
Kawai et al. [15]	✓		
Darabi et al. [9]	✓		
Huang et al. [11]	✓		
He et al. [10]	✓	✓	
Herling et al. [12]		✓	
Iizuka et al. [13]	✓		
Yu et al. [14]		✓	
Isogawa et al. [2, 3](Ch.2)			✓
Isogawa et al. [4](Ch.3)			✓
Isogawa et al. [16](Ch.4)			✓
Isogawa et al. [5, 17](Ch.5)	✓		

For the first problem in this regard, how **different parameters** affect the final result is shown in Fig. 1.2. Fig. 1.2(a) and (b)show the masked image and its inpainted results with different parameters, respectively. Although all inpainted images are generated with the same masked image, their appearances vary largely depending on the inpainting parameters used.

With regard to the second difficulty, Fig. 1.3 shows how **differences in masked regions** affect the inpainting results. Fig. 1.3(a-1) and (a-2) show the original images with masked regions highlighted in red, and (b-1) and (b-2) show the inpainted results of (a-1) and (a-2), respectively. Although there are only slight changes in the masked regions between (a-1) and (a-2), the final inpainted results reveal a significant difference.

Finally, Fig. 1.4 shows a case where the inpainting fails even with the well-tuned parameters and the masked region because of the **inpainting method's limitation**. Fig. 1.4(a) and (b) show the original image with the masked region and its inpainted image. Although the masked region is appropriately drawn to remove the tail of the yellow bird, inpainting fails because there are no hints to fill in the hole, that is, there are no regions that have similar color and texture inside the image.

To solve these three difficulties, although existing inpainting methods focus only on the third part, this thesis also focuses on the first and second difficulties. In Chapter 2 and 3, I propose learning-based preference order between inpainted images and a training data generation method to solve the first issue. To resolve the second difficulty, I propose the method for masked region optimization described

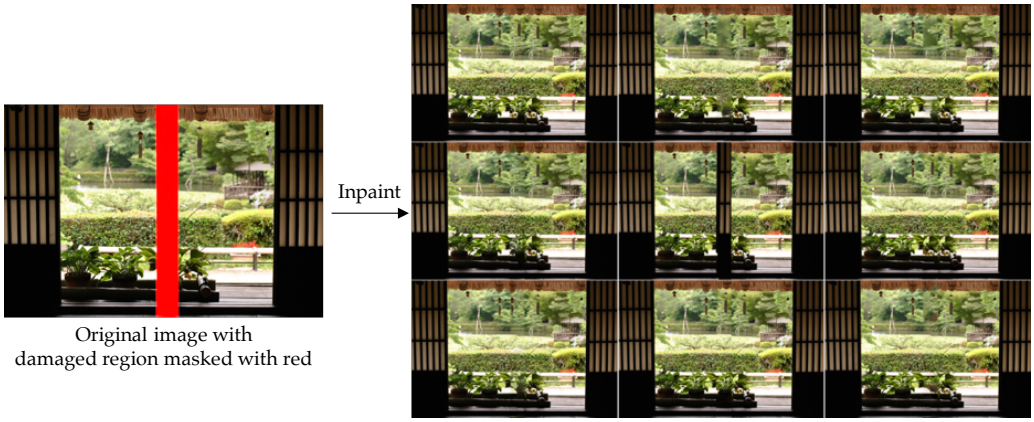


Figure 1.2: Inpainted results with different parameter sets. Although the original image and its masked region are the same as shown in (a), the results vary depending on the parameters used for inpainting as shown in (b).

Figure 1.3: Inpainted results in different masked regions. (a-1) and (a-2) show original images with masked regions highlighted in red, and (b-1) and (b-2) show the inpainted results of (a-1) and (a-2), respectively. Although both results are generated with the same original image, slight changes in the masked regions significantly affect the final image quality.

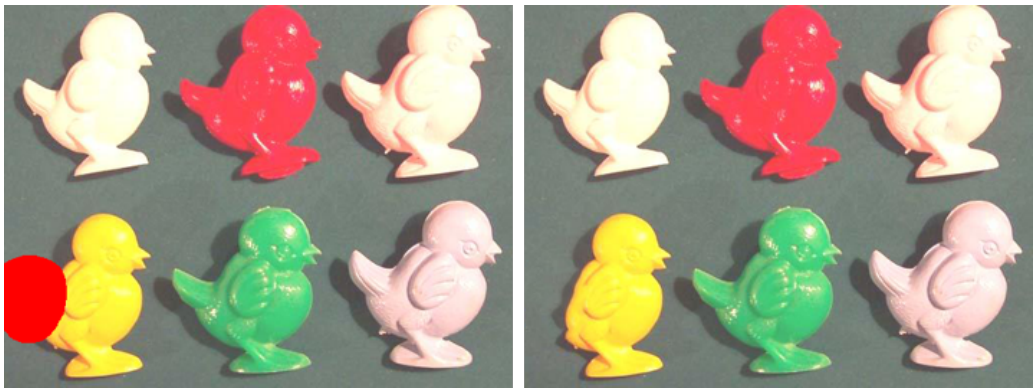


Figure 1.4: Example of inpainting failure due to the lack of similar region. Although same shapes of plastic birds existed, inpainting of the yellow bird's tail fails due to the lack of a similar pattern regarding color and texture.

in Chapter 4. These three chapters were developed to reduce the users' working time associated with trial and error. In particular, in Chapter 5, our method improves the third issue by using a lower dimensional image space for image restoration. The next section describes the contributions of this thesis in greater detail.

1.1 Contributions

The main contribution of this thesis is an exploration of new approaches for improving inpainting techniques in terms of "usability-awareness". This study aims to improve inpainting not only for professional image processing engineers but also for general users. So far, inpainting has been researched to improve restoration itself, and no work has been dedicated to reduce the burden on users. This lack of work for usability with respect to inpainting still requires users' trial and error as described in the previous section, which can be considered as a bottleneck for general users who wish to utilize inpainting. Thus, with this thesis, from the viewpoint of user experience, I have focused on eliminating this bottleneck via four proposed methods. The list of contributions are organized as follows: *preference order estimation, training data generation, masked region optimization, and inpainting via feature reduction and compensation.*

1.1.1 Learning-Based Preference Order Estimation for Image Inpainting

It is widely known that inpainting results vary significantly depending on the method used for inpainting and the parameters set. Thus, in a typical case, users must manually select the inpainting method and the parameters that yield the best result. This manual selection takes a great deal of time; hence, there is a great need for a way to automatically estimate the best result.

To solve the issue, in Chapter 2, I propose an image quality assessment (IQA) method for image inpainting that aims to select the best option from a number of results. Unlike existing IQA methods for inpainting, the proposed method solves this problem as a learning-based ordering task between inpainted images because the problem can be divided into a set of "pairwise preference order estimation" elemental problems. I also introduce effective feature designs enabled, which are investigated by actually measuring human gazes for order estimation.

1.1.2 Training Data Generation without any Manual Operations

Existing learning-based IQA methods for inpainting, such as the approach described in Chapter 2, require subjectively annotated data for training. However, subjective annotation is costly and subjects' judgment occasionally differs from person to person in accordance with the judgment criteria.

To overcome these difficulties, in Chapter 3, I propose a learning-based quality evaluation framework for inpainted results that does not require any subjectively annotated training data. The proposed framework generates and uses simulated failure results of inpainted images whose subjective qualities are controlled as

the training data. I also propose a masking method for generating training datain terms of fully automated training data generation. These approaches make it possible to successfully estimate better inpainted images, even though the task is quite subjective.

1.1.3 Masked Region Optimization for Image Inpainting

In image inpainting, users draw a mask to specify the region. However, it is widely known that users typically need to adjust the masked region by trial and error until they obtain a desired natural inpainting result because inpainting quality is significantly affected by even a slight change in the mask as shown in the previous section. This manual masking takes a significant amount of users' working time and requires considerable input.

To reduce this amount of human labor, Chapter 4 proposes a method for masked-region optimization so that good inpainting results can be automatically obtained. To this end, our approach estimates the "naturalness of inpainting" for all super pixels in inpainted images and reforms an original mask on a super-pixel-basis so that the naturalness of the inpainting result is improved.

1.1.4 Image Inpainting via Feature Reduction and Compensation

Most existing research have failed to perform inpainting processing when similar regions do not exist in undamaged regions. To overcome this, in Chapter 5, a new framework for image inpainting is proposed that creates similar regions by projecting a low-dimensional space from the original space. The approach comprises three stages. First, input images are converted to a lower dimensional feature space. The damaged region is then 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) the possibility of applying patches dissimilar to those in the original color space is enhanced, and (2) the use of many existing restoration methods is made possible, each having various advantages because the feature space for retrieving similar patches is the only extension.

1.2 Outline of Dissertation

The remainder of the thesis is organized in the following manner: **Chapter 2. Learning to Rank Based Preference Order Estimation.** This chapter proposes learning-to-rank based preference order estimation for inpainted images aimed to choose the best result among the several results inpainted with different parameters. The chapter introduces effective image features for learning via eye gaze measurement experiments, and the preference order estimation experiment results suggest the method works with a higher estimation accuracy than existing methods.

Chapter 3. Training Data Generation without any Manual Operations for Learning-Based Preference Order Estimation. Herein, I propose a framework that generates and uses simulated failure results of inpainted images whose subjective qualities are controlled as the training data. I also propose a masking method for generating training data in terms of fully automated training data generation. To demonstrate the effectiveness of our approach, I test our algorithm with various datasets and demonstrate that it outperforms existing IQA methods for inpainting.

Chapter 4. Masked Region Optimization for Image Inpainting. This chapter proposes a new solution to improve image inpainting by optimizing the masked region. To demonstrate the effectiveness of our approach, I test our algorithm with varied images and show that it outperforms existing inpainting methods without masked region reformation.

Chapter 5. Image Inpainting via Dimension Reduction and Compensation. This chapter proposes a new framework for inpainting that uses lower dimensional feature space for restoration. The framework's effectiveness was verified in experiments using various methods, the feature space for restoration in the second stage, and inverse conversion methods.

Chapter 6. Discussion and Conclusion. This chapter summarizes the contributions of this thesis, the remaining issues that need more research, and outlines the future direction of such research.

Chapter 2

Image Quality Assessment for Inpainting via Learning to Rank

This chapter proposes an image quality assessment (IQA) method for image inpainting, aiming at selecting the best one from a plurality of results. It is known that inpainting results vary largely with the method used for inpainting and the parameters set. Thus, in a typical use case, users need to manually select the inpainting method and the parameters that yield the best result. This manual selection takes a great deal of time and thus there is a great need for a way to automatically estimate the best result. Unlike existing IQA methods for inpainting, I and contributors propose the method that solves this problem as a learning-based ordering task between inpainted images. This approach makes it possible to introduce auto-generated training sets for more effective learning, which has been difficult for existing methods because judging inpainting quality is quite subjective. Our method focuses on the following three points: (1) the problem can be divided into a set of "pairwise preference order estimation" elemental problems, (2) this pairwise ordering approach enables a training set to be generated automatically, and (3) effective feature design is enabled by investigating actually measured human gazes for order estimation.

2.1 Introduction

Photos sometimes include unwanted regions such as a person walking in front of a filming target or a trash can on a beautiful beach. Image inpainting is a technique to automatically remove such areas ("damaged regions" in this thesis) and restore them [6, 10, 11, 18, 19, 8, 9, 20]. However, it is known that inpainted results vary largely with the method used and the parameters set[1]. For example, He et al.'s method effectively repairs images including horizontally or vertically repeated textures [10] and Huang et al.'s method is especially efficient for structural images [11]. In the conventional approach to obtain the best inpainting

[1]Inpainting quality is quite substantially affected by parameters such as multi-scale level or patch size to search appropriate regions for restoration. For variations depending on such parameters, please refer to Figs. 2.10 and 2.11.

results, a user selects the inpainting technique and tunes the parameters by trial and error, while observing the inpainted results. Since this is time consuming and requires special knowledge, a method for automatically selecting the best result is required.

Unfortunately, no method has been established to determine the inpainting method and its parameters before conducting inpainting. Instead, we feel that methods for image quality assessment (IQA) that assess the quality of inpainted images have possibility to tackle this purpose. Assessing quality of inpainted images is widely acknowledged as a task that is difficult for automation because its judgment is quite subjective. To handle such tasks, many IQA methods for inpainting focus on how the gaze gathers when a human watches an unnatural image [21, 22, 23, 24, 25, 26, 27]. Through the assumption that unnatural removal of unwanted region gathers human attention, they tried to find a way to represent subjective quality by means of objectively measurable indicators. To obtain human attention, most of these IQA methods use a computational visual saliency map, which simulates human gaze density [22, 24, 25, 26, 27].

Although the basic idea of using human attention is reasonable, these existing methods are difficult to apply for comparing the qualities of two inpainted images due to the following two factors. One is the difficulty in estimating human attention. Actual human attention changes by contexts such as the reason for viewing. Isogawa et al. [3] revealed that the human gaze pattern while watching inpainted images is different from any computational saliency maps. The other one is the resolution of a saliency map, which is generally coarse. Thus, it is difficult to apply this method to the current task in which the difference resides in a locally ubiquitous way.

Estimation of subjective quality is not unique to inpainting image. In the research field of subjective-evaluation-estimation, learning to rank approaches have been investigated actively [28, 29, 30, 31, 32, 33, 34, 35]. Although for learning based approaches, large and representative training data sets are essential to improve estimation accuracy, accumulating training data is difficult in view of annotation cost and fluctuation of user annotation. The learning to rank framework replaces the subjective evaluation tasks as an ordering task without estimating an absolute score. It is considered that it opens up a new era within the research field of subjective evaluation. The problem setting is quite reasonable and reduces preparation costs since it enables learning without absolute scores annotated by human subjects; selecting the better one is rather easier than providing the scores for images to be subjectively evaluated such as inpainted images. We consider that the learning to rank framework has the potential to further reduce the training data accumulation by making good use of ordering traits. Recently, some studies have automatically generated and/or augmented training data by image processing [36, 37, 38]. We believe this concept is also applicable to IQA for inpainting.

In this chapter, we show how we tackle the task of obtaining the best inpainted result among inpainted images obtained through various methods and parameters.

We also propose a new learning to rank based ordering approach for inpainted images. Unlike existing IQA methods for inpainting, our method has a new feature that does not use a computational visual saliency map but uses our investigation of human gaze while watching inpainted images. Another important proposal is automatic generation of training data. By making good use of pair-wise learning, we propose automatic generation of training pairs to improve estimation accuracy. The contributions of this chapter are as follows:

- This is the first trial for applying learning to rank for IQA of inpainted images.
- The proposed method enables automatically generated training data to be introduced by making good use of a ranking mechanism, although the learning target is quite subjective.
- It proposes new image features dedicated to inpainted image quality assessment on the basis of gaze measurement experiments.

This chapter is based on our previous conference proceedings [3] and adds a comprehensive investigation on how the proposed features work and a novel method for accumulating training data automatically to improve estimation accuracy. The rest of this chapter is organized as follows. In Section **2.2** we briefly review related work. Section **2.3** investigates actual human gazes to design effective image features for learning. Section **2.4** describes the learning based ranking method we propose, which was developed with the knowledge detailed in Section **2.3**. In Section **2.5** we verify the method's effectiveness by comparing it with existing IQA methods. We also describe the effectiveness of introducing an auto-generated training set. In Section **2.6**, we conclude the chapter with a summary of key points and mention future work to be done.

2.2 Related Work

This section reviews previous studies. Subsection 2.2.1 introduces IQA methods for image inpainting whose purposes are the same as ours. Then, in 2.2.2, we introduce a learning to rank approach that has attracted attention as a method for estimating subjective evaluations.

2.2.1 IQA Methods for Image Inpainting

Estimating quality is one of the difficult issues for image inpainting. The main reasons are the ambiguity in subjective evaluations, and the cost for obtaining training data. Because of the former issue, although many effective IQA methods for degraded images, e.g., burred, compressed, or noised images have been proposed [39, 32, 33, 34, 35], these methods cannot be applied assessing inpainted images.

To overcome the former issue and to obtain subjective evaluations stably, there are three main approaches: reflecting human reactions such as gaze transition [23, 24, 25], asking subjects to provide their judgments [26], and combining them [27].

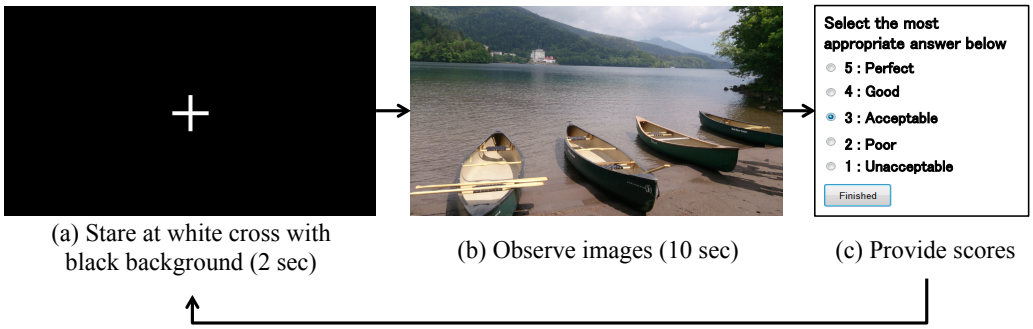
The basic concept of human reaction based IQA is that less natural inpainted regions will gather more gazes because of the unnaturalness for human perception. Thus, this method estimates inpainted image quality on the basis of gaze density before and after images are inpainted [23]. To reduce the cost for measuring actual human gazes, many metrics use computational visual saliency maps instead of actual gazes [24, 25, 27]. A computational visual saliency map ("saliency map" for short), is a topographically arranged map that represents estimated visual saliency only from the image. If saliency maps well reflect actual human gaze patterns, substituting them for actual gazes will work well. However, the accuracy of saliency maps is unfortunately quite limited as we mention in Section 2.3, and thus the performance of saliency map-based IQA methods is also limited.

Learning based approaches that depend on support vector regression (SVR) have also been reported [26, 27]. For these methods, subjectively annotated rating scores are essential for training regression models. Because of the need for absolutely subjective scores, all of the training data should be manually annotated, which is the second issue for building an IQA method for inpainting. Thus generating training sets requires quite high annotation cost. To overcome this issue, our method generates training data automatically as described in 2.4.2.

2.2.2 Ranking based Image Evaluation for Subjective Judgment

In many subjective evaluation tasks, it is difficult to provide absolute scores. For example, scoring the degree of smiles is a quite difficult task and the scores may vary largely by question. Since estimating such varied subjective scores is quite difficult, sidestep methods have been widely considered. learning to rank based approaches are now acknowledged as a promising solution. Rather than absolute scores, they provide a learning framework for merely ordering scores among target samples. Coming back to the above cited example, sorting the images by degree of smile is easier than giving smile scores to each image.

Among learning to rank approach variants, pairwise learning to rank methods have gathered attention due to the ease with which they can be implemented. They have been frequently applied for estimating preference order [28, 29, 30, 31]. Chang et al. estimated the age of a single face image [28]. Yan et al. obtained the most visually appealing color enhancement of an image [29]. Abe et al. estimated the surface qualities of an object, such as glossiness or transparency from its images [30], and Khosla et al. estimated the most memorable region inside images [31].



(a) Stare at white cross with black background (2 sec)

(b) Observe images (10 sec)

(c) Provide scores

Figure 2.1: Three step test procedure of the preliminary experiment conducted to elucidate the relationship between human gaze and subjective scores. Subjects are required to; (a) stare at the white cross to fix initial viewpoint, (b) observe an image, and (c) provide a 5-point opinion score (subjective assessment) of the image.

learning to rank has also been introduced in IQA methods [32, 33, 34, 35]. Gao et al. [32] and Xu et al. [33, 34] proposed blind image quality assessment frameworks for degraded images, e.g., blurred or compressed images or images with white noise. In addition, Ma et al. introduced learning to rank to assessing retargeted images [35]. Although retargeted images are quite different from general degraded images that existing IQA methods deal with, they examined and investigated the effects of learning to rank based IQA for image retargeting.

Unlike existing methods, the method discussed in this chapter focuses on assessing inpainted images. Since estimating the quality of inpainted images is a quite different task than assessing other deteriorated images, we designed new image features dedicated for assessing inpainted images. In addition, this chapter shows that by using pairwise learning traits we can produce training data automatically and use the data to improve estimation accuracy.

2.3 Toward Effective Image Features: Eye Gaze Investigation

Many IQA methods use visual saliency maps as substitutes for actual gazes. However, we have doubts about the coherence of computational visual saliency and actual human gazes, especially when observing inpainted images. Therefore, before we go into the proposed method, we will describe an eye gaze measurement experiment we conducted for two purposes. The first was to show the difference between measured gazes and the saliency map and to reveal the difficulty in using saliency maps instead of actual human gazes for IQA. The second was to analyze the region and features within inpainted images we should focus on to assess the quality of the images on the basis of measured gazes and the corresponding subjective evaluations.

2.3.1 Procedure and Set-Up of Eye Tracking Experiment

We conducted this experiment with the aim of verifying coherence between the measured gazes and the saliency map. We also obtained subjective scores for each image to investigate how gazes affected the total subjective image quality. Fig. 2.1 shows the test procedure, in which subjects repeated three tasks: (a) stare at a white cross on a black background for two seconds to fix their initial viewpoint, (b) observe images for 10 seconds, and (c) provide 5-point opinion scores representing image quality unnaturalness. The scores 1-5 respectively corresponded to *Very noticeable, Rather noticeable, Slightly noticeable, Hardly noticeable, and Unnoticeable*. Higher scores are better since they indicate that the unnaturalness that occurs with inpainting is unnoticeable.

The observed image in task (b) includes original images and inpainted images. Original images are images that are not inpainted. Inpainted images were generated with two methods, i.e., those reported by He et al. [10] and Huang et al. [11]. The subjects had no prior knowledge on the types of images displayed (i.e., whether they were original images or images that had been inpainted by using the methods reported by He et al. [10] and Huang et al. [11]).

To prevent the subjects from having prior knowledge of the material, we generated three types of images generated from each of 100 original images. We asked 24 subjects (8 males and 16 females) with normal vision to report the image quality after observing the displayed images. These subjects were divided into three groups and the subjects in each group watched the same type of image. Each subject watched 100 images.

We applied a stationary Tobii eye tracker for gaze measurement. The LCD monitor used for stimulus presentation was 21 inches (1280×1080 pixels). The monitor-observer distance was 60 cm.

2.3.2 Integrity between Computational Saliency Maps and Human Visual Attention

In Fig. 2.2, (a) shows an inpainting target image and (b) shows the inpainted result in which an undesired man standing in front of a boat was removed. Measured human attention is overlaid in (c). Calculated visual saliency maps obtained with calculation methods proposed by Hou et al. [40], Achanta et al.'s [41], and Walther et al.'s [42] are respectively shown in (d), (e), and (f). These maps have actually been used for assessing the image quality of inpainted images; they were respectively used by Voronin et al. [27], Trung et al. [25], and Oncu et al. [24]. From these maps, we can observe that the resolution of computational visual saliency maps is quite coarse and their results are significantly varied. Additionally, saliency maps are quite different from human visual attention. As shown in Fig. 2.2(b), inpainting failed to fill the shape of the boat. Because this failure produces significant unnaturalness, the most salient areas for actual human gazes

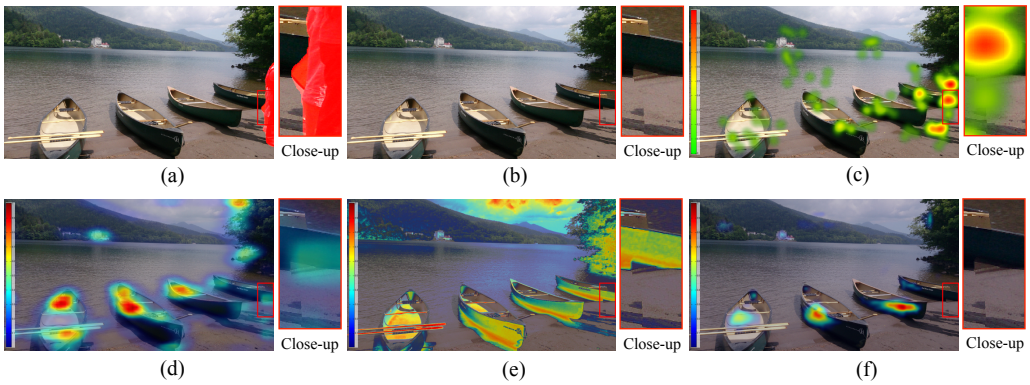


Figure 2.2: Comparison between observed human visual attention and computational visual saliency. (a) Inpainting target image. (b) Inpainted image. (c) Human visual attention overlaid on (b) (red gathers more gazes). (e)-(f) Computational visual saliency overlaid on (b) (red gathers more gazes). Saliency maps are (d) Hou et al.'s [40] used in Voronin et al.'s metric[27], (e) Achanta et al.'s [41] used in Trung et al.'s metric [25], and (f) Walther et al.'s [42] used in Oncu et al.'s metric [24].

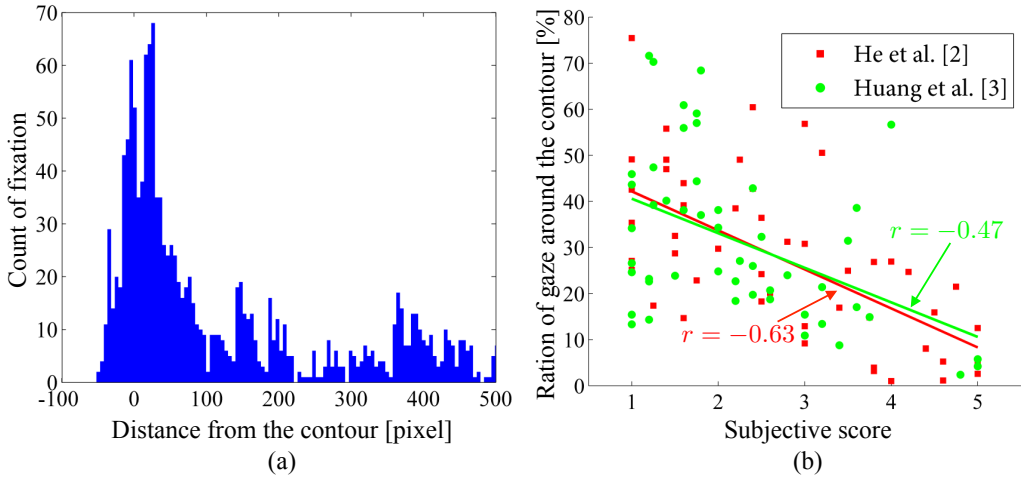


Figure 2.3: Gaze measurement results; (a) gaze histogram with Fig. 2.2(b), (b) relationship between the subjective score and density of gaze within the vicinity of damaged region's contour.

were those around the damaged boat (See Fig. 2.2(c)). The areas around the boat in (d) and (e) were somewhat salient, but were more salient in other areas (e.g., the other boats or the oars). In (f) no saliency around the boat was represented at all. These results suggest that it is difficult to use computational visual saliency maps as a substitute for human gazes. Thus, it is essential to come up with new image features that represent such unnaturalness.

2.3.3 Correlation between Subjective Quality and Human Visual Attention to Damaged Region Contours

Human gazes are potentially an excellent means for assessing inpainting quality, and a metric based on human gazes was proposed by Venkatesh et al. [23]. This metric categorizes eye gaze position into two categories, i.e., inside and outside damaged regions. They use the difference of amount of gazes between pre- and post- inpainted images. We were inspired by this simple and effective idea and so tackled further analysis of eye gaze patterns in categories other than inside and outside damaged regions. This section shows how we analyzed what we should focus on to assess the quality of the inpainted images on the basis of knowledge of human attention and corresponding subjective evaluations. We believe that this knowledge will be useful in developing an IQA method for image inpainting.

We analyzed the characteristics of observed gaze and corresponding MOS levels. The MOS values are the average of the 5-point annotated scores provided by subjects as described in the previous section. We first investigated on where human beings tend to watch for inpainted results. Fig. 2.3(a) shows an example gaze histogram for the inpainted image in Fig. 2.2(b). Its vertical axis is the time the gaze was oriented and the horizontal axis is the distance from the contour of a damaged region, where a negative value means inside the damaged region. As shown in the histogram, around the contour, i.e., distance $= 0$, gathers more gazes, which indicates the contour of the damaged region tends to be salient.

To be more specific, Fig. 2.3(b) shows the relationship between subjective score and gaze density within the vicinity of a damaged region's contour for two different inpainting methods. Green and red points are for the methods proposed by He et al. [10] and Huang et al. [11]. Here, we set the contour vicinity to be within 30 pixels from the contour. This corresponds to the 1.0 degree view angle we used in our experimental setup. As shown in Fig. 2.3(b), the correlation coefficients r for the inpainted results provided by He et al. and Huang et al. are respectively $r = -0.63$ and $r = -0.47$. These results indicate a high negative correlation exists between subjective scores and gaze density around the contour. Thus there is a high probability that the image features around the contour are important for preference estimation.

2.4 Proposed Method

Now we are ready to describe our proposed method. Our goal is to obtain the best result among inpainted images that are generated with different inpainting methods and parameters. In 2.4.1, we describe our pairwise ordering method and in 2.4.2 show how we automatically generated large auto-generated training set in which manual intervention was not required. Then, in 2.4.3 we show effective image features for this approach, which include the knowledge given in Section 2.3.

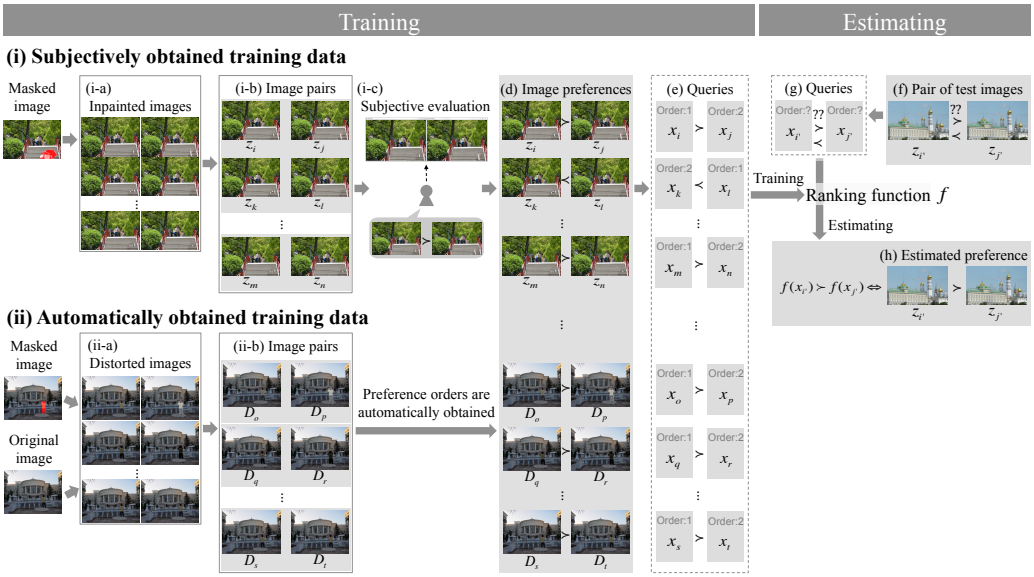


Figure 2.4: Proposed method overview.

2.4.1 Ranking by Assessing Image Quality with Learning

Aiming at tackling the difficulty in reflecting subjective evaluations for inpainted results to scores, we based our preference order estimation on a learning to rank approach. Fig. 2.4 shows the overview of our proposed method.

Before we explain the details, let us briefly explain a typical pairwise learning to rank algorithm. This algorithm premises a ranking function $f(x)$, which computes the strength of the target attribute for each sample, as described below. Hereafter, we use x_i to denote a feature vector extracted from sample image z_i. The $f(x)$ is trained so that the ordering of the output value from the function $f(x)$ reflects the user annotated preference order $z_i \succ z_j$ between image pairs. In a word, the function f should satisfy the following formula:

$$z_i \succ z_j \iff f(x_i) > f(x_j) \tag{2.1}$$

We modeled f with the linear function $f(x) = \omega^\top x$. Then inequalities (4.1) can be written as below.

$$z_i \succ z_j \iff \omega^\top (x_i - x_j) > 0 \tag{2.2}$$

This mirrors the problem of binary classification. To implement the formulation that uses binary classification to calculate the preference order of pairs of images, the pair-wise learning to rank approach is widely used. From among the various methods yielding pair-wise learning to rank, we adopted RankingSVM [43] as it is used widely [29, 30, 31] due to its effectiveness and ease of implementation.

In our method, function f is trained with the pair of image feature vectors described in 2.4.3 with their preferences. We call a training data set of this type

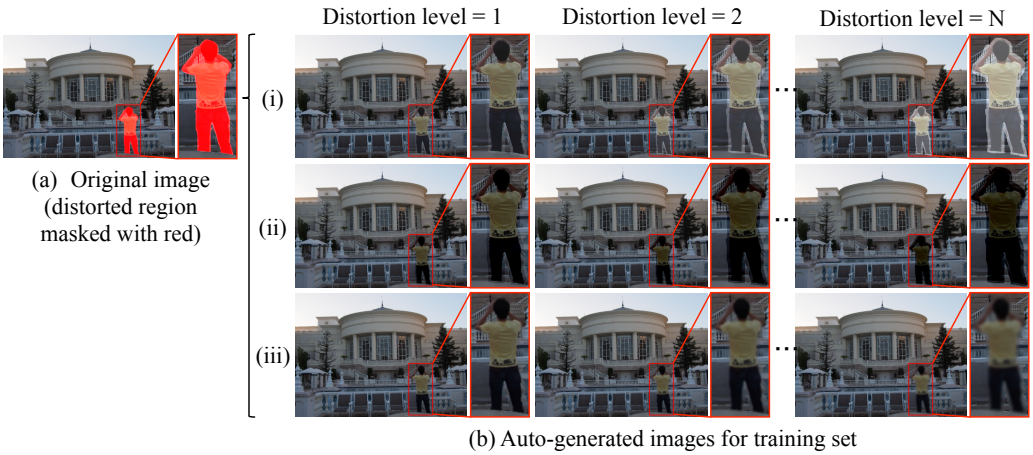


Figure 2.5: Auto-generated images for training set. (a) Original image with region to be synthesized, which is masked in red. (b) Multi-levels of auto-generated images with (i) brighter color distortion, (ii) darker color distortion, (iii) blurred distortion.

a "query" (see Fig. 2.4(e)). Basically, all preferences for generating queries are manually annotated by subjects. First, inpainted images with several parameters are generated with a masked image as shown in Fig. 2.4(i-a). These results are used to make a pair of train images like those in Fig. 2.4(i-b). Then, as shown in Fig. 2.4(i-c), subjects were asked to provide their preference judgment (x) to a pair of inpainted images (z). Subjects' preferences are reflected to all image pairs (Fig. 2.4(i-d)).

However, the training data shortage problem still remains. To solve this problem, we additionally propose a way to automatically generate training data as shown in Fig. 2.4(ii). In this case, auto-generated images (Fig. 2.4(ii-a)) are used instead of inpainted images. The difference is that we already know that preferences depend on the degradation level. Thus we can skip to annotation; images with preferences (Fig. 2.4(ii-d)) are directly generated with image pairs (Fig. 2.4(ii-b)). In the next section we will describe how we designed a way to automatically generate a training set for which manual annotation is not required.

2.4.2 Auto-Generated Training Data

Existing learning-based IQA methods for inpainting learn the relationship between image features and corresponding scores provided by subjects. Thus, they require user annotated samples. Here, we propose one effective solution, i.e., an IQA method for inpainted images by pairwise ordering. It is effective because it does not need any absolute scores but only pairwise relationships.

We add some distortions, such as proportional changes in pixel values or applying a low pass filter, to the original images that tend to occur as the result of inpainting. Several levels of such distorted images and original images generate

training data with the assumption that increased distortion lessens preference. Of course, the original image has better quality than the distorted image. Because our method requires only pairwise relationships, not absolute scores, this simple relationship in which images become more distorted can work as a training data source. Fig. 2.5 shows examples of several levels of auto-generated images for training. The i-th auto-generated train image I_i is synthesized by combining the original image I_{orig} and the i-th distorted image D_i as below.

$$I_i(x,y) = \begin{cases} D_i^{\gamma}(x,y) & ((x,y) \in \Omega) \\ I_{orig}(x,y) & (otherwise) \end{cases} \tag{2.3}$$

$$\gamma = \begin{cases} c & (brighter\,or\,darker\,color\,distortion) \\ b & (blur\,distortion) \end{cases} \tag{2.4}$$

where Ω is a masked region to reflect distortion (see Fig. 2.5(a)).

The reason we apply color and blur distortion is to simulate typical failures occurring as a result of inpainting. Human attention is considered to be quite sensitive to unnaturalness that is produced by differences in brightness and frequency components in images. To represent unnaturalness of this type, we focus on three types of distortion, i.e., that caused by brighter colors, darker colors, and blurring. The former two distortions represent undesired inpainted results in which the colors of inpainted regions are brighter or darker color than those outside the contour of the inpainted region. The latter represents undesired inpainted results in which there is edge discontinuity around the inpainted region. Distorted images of these types are generated as follows.

Brighter and darker color distorted images
We define the i-th brighter/darker damaged image as below:

$$D_i^{(c)}(x,y) = I_{orig}(x,y) + \alpha\,\beta\,i \tag{2.5}$$

$$\beta = \begin{cases} +1 & (brighter\,color\,distortion) \\ -1 & (darker\,color\,distortion) \end{cases} \tag{2.6}$$

where α is a scalar parameter. In the work we report in this chapter, we set $\alpha = 10$.

Blurred distorted image
We define the i-th blurred distorted image as below:

$$D_i^{(b)}(x,y) = \sum_{(k,l)} I_{orig}(x-k, y-l)\, G(k,l) \tag{2.7}$$

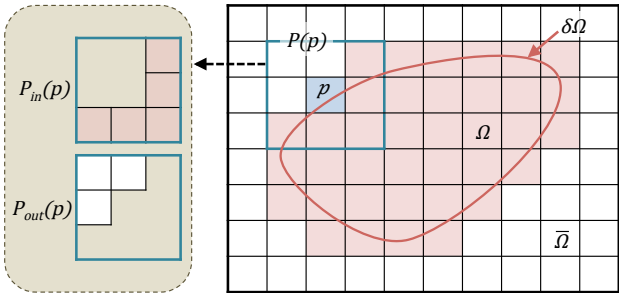


Figure 2.6: Damaged/distorted region and its contour. $P_{in}(p)$ and $P_{out}(p)$ show masked or non masked regions in patch $P(p)$, which is centered at point p.

Here, G is a 2-dimensional Gaussian distribution with kernel size γ and we set $\gamma = 2i+1$.

2.4.3 Features for Learning to Rank

Using the observation provided in Section 2.3, we designed image features for our framework. We call this image feature patch-based contour consistency (*PBCC*). As we described in Section 2.3, human perception is quite sensitive to color or edge discontinuities between in/out of damaged/distorted regions, which we combined to design the *PBCC*.

The *PBCC* consists of the following two components: (1) differences between in/out of damaged/distorted region, and (2) normalized image around the contour. The former represents continuity across the contour of the damaged/distorted region. The latter represents the relative quality of images; the coherence of image quality between the inside and outside parts of a damaged region largely affects subjective quality. Thus, even if the image quality within the damaged region is the same, its perceptive quality varies depending on its surrounding region's quality.

To make the features dedicated for evaluating inpainted images, these components are computed along contours of the damaged/distorted region as shown in Fig. 2.6. We set the features as $x = (X_d, X_s)$, where X_d and X_s respectively represent the first and the second components. X_d and X_s are computed as below;

$$X_d = ||S(P_{in}) - S(P_{out})||_2^2 \tag{2.8}$$

$$X_s = \frac{\sum_{p\in\delta\Omega} S(P_{out}(p))}{\sum_{p\in\delta\Omega} 1} \tag{2.9}$$

where Ω and $\delta\Omega$ respectively denote a damaged/distorted region and its contour. Eq. A.1 represents squared 2-norm. $P_{in}(p)$ and $P_{out}(p)$ show masked or non

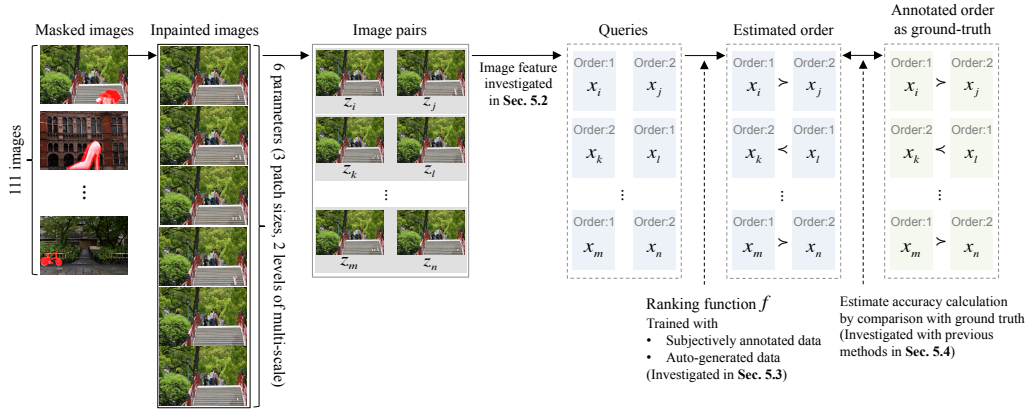


Figure 2.7: The flowchart for experiments conducted in Section 2.5. In Section 5.2, we investigated the efficacy of our proposed image representation. Section 5.3 verifies the effectiveness of auto-generated training data and Section 5.4 compares the estimation accuracy of our method and previous methods to assess their performance.

masked regions in patch $P(p)$, which is centered at point p (See Fig. 2.6). In addition, $S(P_{in}(p))$ and $S(P_{out}(p))$ represent average features of $P_{in}(p)$ and $P_{out}(p)$ as shown below.

$$S(P_{in}(p)) = \frac{\sum_{q \in P(p) \cap \Omega} s(q)}{\sum_{q \in P(p) \cap \Omega} 1} \tag{2.10}$$

$$S(P_{out}(p)) = \frac{\sum_{q \in P(p) \cap \bar{\Omega}} s(q)}{\sum_{q \in P(p) \cap \bar{\Omega}} 1} \tag{2.11}$$

To the extent of the work we report in this chapter, we used $s(p) = (u(p), v(p))$, where $u(p) = (u_R(p), u_G(p), u_B(p))$ and $v(p)$, each denoting RGB pixel values and edge strength.

2.5 Experiments

This section describes how we investigated the effectiveness of the proposed method. We will start by detailing the experimental setups used in 2.5.1. We will then verify the effectiveness our method by using the image features we derived and auto-generated training data as described in 2.5.2 and 2.5.3, respectively. 2.5.4 then compares our method with the other existing IQA methods. For easy understanding, Fig. 2.7 shows the flow chart of the experiments conducted in this section.

Figure 2.8: Annotation interface for obtaining training data. Two different inpainted results are displayed side by side. Subjects annotate their preferences among three options: **r**: right image is better, **l**: left image is better, and **n**: no preference order.

2.5.1 Experimental Setup

To generate training and test images, 111 images with manually masked damaged regions were prepared. The 111 images were inpainted with two inpainting methods [10, 11] and six parameters (3 patch sizes and 2 levels of multi-scale parameters) were used for each method. The quality of the images was evaluated by 24 subjects (12 males and 12 females) with normal vision. To make the users' judgment easy, we randomly displayed a pair of inpainted images side-by-side as shown in Fig. 2.8. Subjects were asked to choose one of three options: **r**: right image is better, **l**: left image is better, and **n**: no preference order (i.e., it is hard to decide which one is better or which one is worse). Excluding inpainted images with extremely poor quality, we obtained 2,466 image pairs. We excluded poor quality images because they might change subjects' judgement criteria during the experiment.

We implemented RankingSVM with SVM Rank [44] with Radial Basis Function (RBF) as the kernel function ($\gamma = 2^{-7}$), and the regularization parameter ($C = 2^{-5}$). We used a desktop PC (Intel Core i7, 3.4GHz CPU, 32GB memory) and used Matlab to implement the existing method.

2.5.2 Performance Comparisons for Different Image Features

This subsection verifies the effectiveness of *PBCC*, the proposed image feature described in 2.4.3. Table 2.1 compares the performances attained with seven different image features: F_{all} [32], GIST [45], EMD kernel [26, 27], $Saliency$, F_{in}, F_{out}, and *PBCC*. Note that in verifying performance with these image features we used the same training data and estimator; only the image features were different.

Here, we briefly introduce each of the compared features. F_{all} was originally used for learning-based-IQA of degraded images. GIST is one of the most commonly used global image features and is used for learning-based IQA for retargeted images. The EMD kernel is calculated on the basis of EMD and has been used in

Table 2.1: Performance comparison for different image features. The highest score is underlined. [%]

Feature	estimation accuracy
F_{all} [32]	44.47
GIST [45]	60.80
EMD kernel [26, 27]	53.27
Saliency [42]	57.03
F_{in}	45.73
F_{out}	40.45
PBCC(Ours)	<u>70.10</u>

learning-based-IQA for inpainted images. F_{in} and F_{out} are the original features of this chapter and represent the inside and outside parts of inpainted regions. The two methods represented by Eq.12 and 13. were used for verifying the effectiveness of contour consistency on which *PBCC* focuses. *Saliency* is for comparison of computational saliency maps. The same as *PBCC*, *Saliency* is calculated so that it represents the inside and outside parts of inpainted regions and depends on how *PBCC* is claculated. We used Walther et al.'s computational saliency map [42], as it is used by previous work [24].

$$F_{in} = \frac{\sum_{q \cap \Omega} s(q)}{\sum_{q \cap \Omega} 1} \tag{2.12}$$

$$F_{out} = \frac{\sum_{q \cap \bar{\Omega}} s(q)}{\sum_{q \cap \bar{\Omega}} 1} \tag{2.13}$$

Table 2.1 shows the estimation accuracy obtained, which is the ratio at which the preference order is correctly estimated among annotated pairs. As can be seen from the table, *PBCC* correctly estimated the image pair preferences at 70.10%, as opposed to the 40.45% to 60.80% obtained with other methods. These results confirm the effectiveness of the proposed *PBCC*.

2.5.3 Verification of Effectiveness Depends on the Amount of Auto-Generated Training Data

In this section we will show the results obtained in an investigation we conducted, which indicate that the system performance changes depending on the size of the auto-generated training set. First, we will show the results obtained with $N = 0$, in which no auto-generated images are included. Second, we will show how an auto-generated training set affects the performance. We used 100 original images

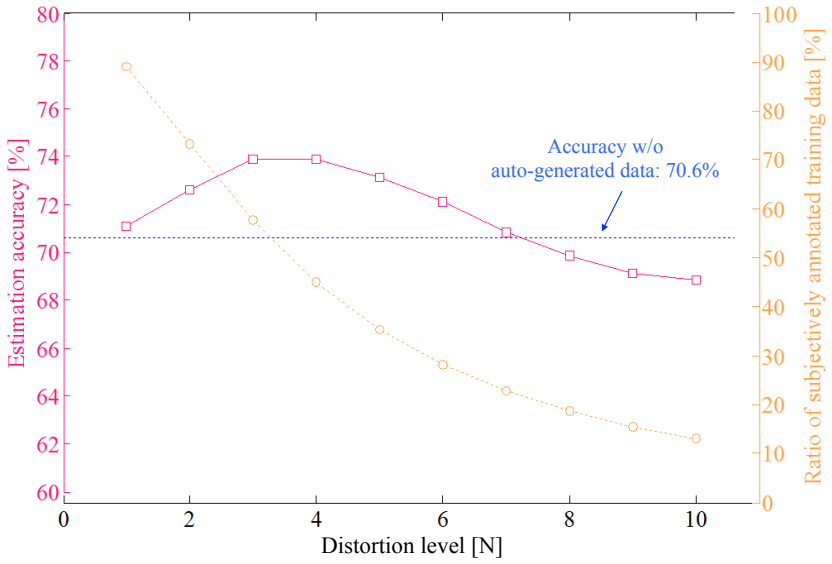


Figure 2.9: Investigation of performance depending on the amount of auto-generated training set. Estimation accuracy depending on distortion level N is shown with magenta line with left y-axis. Accuracy without any auto-generated data is shown in blue line. Ratio of subjectively annotated training data to whole data is shown in orange bar graph with right y-axis.

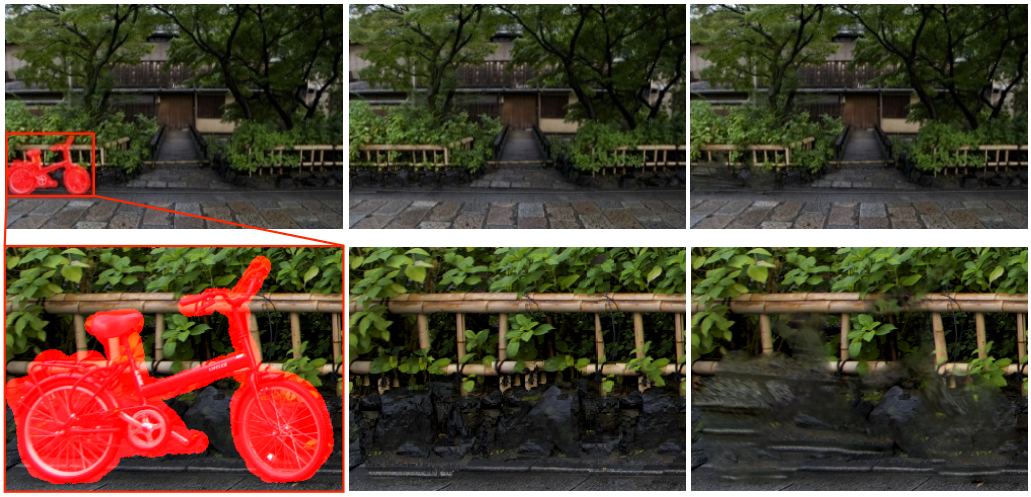
to generate auto-generated image levels of $N = 1$ to 10. Each level included the three types of distortion described in 2.4.2.

N levels of auto-generated images make ${}_{N+1}C_2$ combinations of pair images, including comparisons with the original image. All of these ${}_{N+1}C_2$ pairs are used for training. We investigated the effect of the amount of data with $N = 1$ to 10. Fig. 2.9 shows a line graph in which the left y-axis shows estimation accuracy depending on N, which means the number of auto-generated images. Estimation accuracy without any auto-generated training set is annotated with a blue line for reference. The ratio of subjectively annotated training data to the data as a whole is shown by the orange line for the right y-axis. The training set in which N was 3 consists of 900 pairs of samples based on the 100 original images, with three types and three levels (six distortion combinations). In the same way, the training set in which N was 4 consists of 1800 pairs based on the 100 original images, with three types and four levels (10 distortion combinations). These results indicate that a larger training set is more effective; however, a set larger than certain levels of images results in worse performance.

One of the possible causes for this is that training sets with similar data may result in worse total prediction accuracy. If N is increased, a similar training set generated with the same original images will also increase. This additional data may decrease the effectiveness of learning. We also consider that another cause may be that the rate of subjective annotated data decreases as the amount of auto-generated data becomes larger. In the next section, we will show how we used two settings with different amounts of auto-generated data as our proposed methods:

Table 2.2: Prediction accuracy comparison with existing image quality assessment metrics. The highest score is underlined. [%]

Method	estimation accuracy
ASVS [22]	58.04
DN [22]	60.22
$\overline{GD_{in}}$ [23]	55.88
BorSal [24]	50.46
StructBorSal [24]	51.55
Ours (w/o auto-generated data)	70.10
Ours (w/ auto-generated data)	<u>73.87</u>



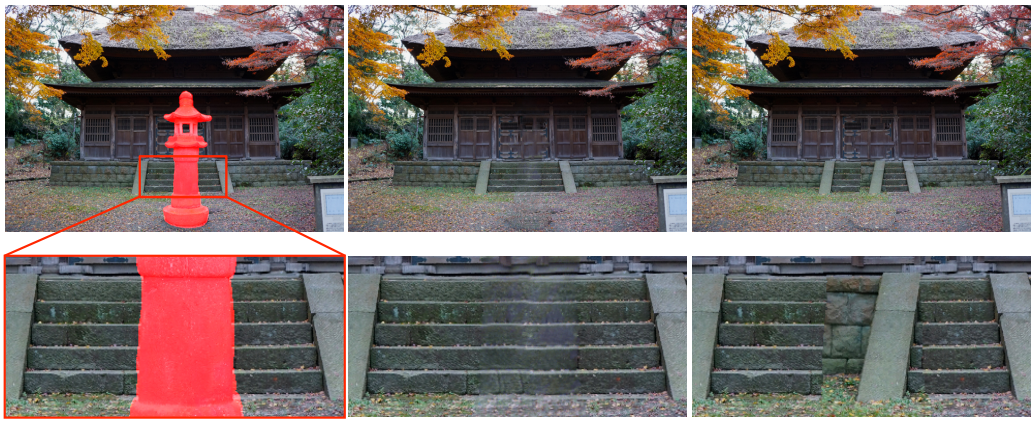
(a) (b) (c)

Figure 2.10: Correctly ordered images with proposed method; (a) original image with damaged region masked in red while (b) and (c) are inpainted pairs of images that subjects annotated as $(b) \succ (c)$.

$N = 0$ with no auto-generated images, and $N = 3$, which was most effective for learning.

2.5.4 Comparison with Existing Methods

We conducted experiments in which we compared our method with other IQA methods for image inpainting, i.e., *ASVS* and *DN* by Ardis et al. [22], $\overline{GD_{in}}$ by Venkatesh et al. [23], and $BorSal, StructBorSal$ by Oncu et al. [24]. Because we did not use an actual eye gazes for this experiment, we used a saliency map instead of human gaze for $\overline{GD_{in}}$; this is the same method that was used in the comparison experiment reported by Oncu et al. [24]. For our method we used $N = 0$ without an auto-generated training set and $N = 3$ with such a set; the results obtained were presented in 2.5.3.



(a) (b) (c)

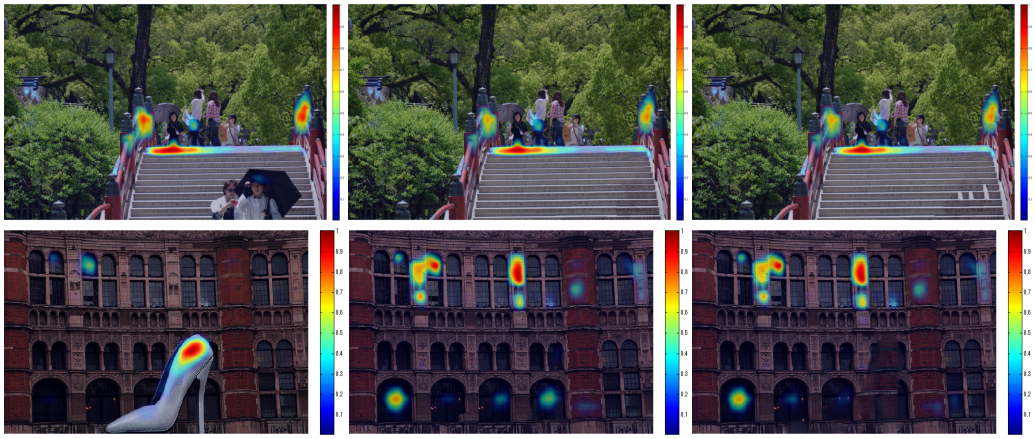
Figure 2.11: Incorrectly ordered images with proposed method; (a) original image with damaged region masked in red while (b) and (c) are inpainted pairs of images that subjects annotated as $(b) \succ (c)$.

Tab. 2.2 shows the prediction accuracy obtained for each metric. Our method without/with auto-generated training data correctly estimated the image pair preferences at 70.10% and 73.87% respectively, as opposed to the 50.46% to 60.22% obtained with other metrics. Thus, the improvement our method achieved over existing methods was around 13 percentage points.

Figs. 2.10 and 2.11 respectively show example outputs of correct and incorrect estimation obtained with our method. In both figures, (a) is an original image with a damaged region marked in red, and (b) and (c) are the inpainted results annotated by subjects as $(b) \succ (c)$. In Fig. 2.10, (b) images are successfully inpainted while example (c) was an inpainting failure in terms of both color consistency and structure. Our method successfully estimates preferences for such image pairs. We consider that because our image feature design focuses on the unnaturalness produced by color or structural discontinuity, it works well as the Fig. 2.10 results show. It is especially notable that none of the other existing metrics correctly estimated the preferences for the top left images in Tab. 2.2. Other than StructBorSal [24], these methods also failed to estimate preferences for the bottom left images.

We consider that these methods failed due to the uncertainty of the computed visual saliency maps. To demonstrate the cause of the previous method's failure, Fig. 2.12 shows a saliency map overlaid on the left half images of Fig. 2.10. In Fig. 2.12, (a) to (c) correspond to (a) to (c) in Fig. 2.10; the original image and inpainted pairs of images. All subjects answered that (b) was better than (c).

The upper (c) in Fig. 2.12 includes an inpainting failure around the stairs. However, neither (b) nor (c) gather saliency around the inpainted region. Also, the lower (c) in Fig. 2.12 has color and structural discontinuity, which generates huge unnaturalness. However (b) gathers more saliency on un-inpainted regions. We consider that this type of uncertainty and instability in saliency maps impede IQA quality.



(a) (b) (c)

Figure 2.12: To show the cause of the other existing methods' failure, a saliency map is overlaid on the left top and bottom images in Fig. 2.10. $(a) - (c)$ are related to Fig. 2.10; original image and inpainted images. Upper images show that there are no significant differences between the two inpainted images. In the lower images, (b) gathered more gazes although subjects preferred (b).

Our method's limitation is shown in Fig. 2.11. In this figure, (b) has a blurred region and also has color discontinuity, but it is relatively natural in context. In (c), if we hide the left half of the picture, it is quite natural. However, structural unnaturalness occurs in the left half of the image and generates context unnaturalness. Because human perception is considered to be more sensitive for contextual failures, subjects preferred (b). Currently our method does not consider any semantic information, thus for such image pairs it generates ordering failures.

2.6 Conclusion

In this chapter we described an image quality assessment method we developed for image inpainting. Three key ideas of our method are that (1) we use a ranking-by-learning algorithm to estimate the ordering of inpainted images on the basis of subjective quality, (2) our ranking system easily introduces auto-generated training data for more effective learning, and (3) we introducing image features that reflect differences around a contour of damaged regions on the basis of gaze measuring experiments which showed that a high negative correlation exists between subjective quality and gaze density around the contour. Unlike existing image quality assessment (IQA) methods for image inpainting, ours makes it possible to introduce auto-generated training data, due to introduction of our pairwise learning. Preference order estimation experiment results suggest the method's efficacy. Especially with auto-generated training sets, the estimation performance was about 13 percentage points higher than that of existing IQA methods. In future work, we will introduce other image features such as describing semantic unnaturalness inside inpainted images.

Chapter 3

Training Data Generation without Any Manual Operations

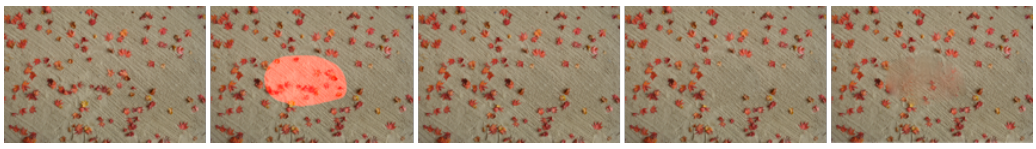
In this chapter I and contributors propose a learning-based quality evaluation framework for inpainted results that does not require any subjectively annotated training data. Image inpainting, which removes and restores unwanted regions in images, is widely acknowledged as a task whose results are quite difficult to evaluate objectively. Thus, existing learning-based image quality assessment (IQA) methods for inpainting require subjectively annotated data for training. However, subjective annotation requires huge cost and subjects' judgment occasionally differs from person to person in accordance with the judgment criteria. To overcome these difficulties, the proposed framework generates and uses simulated failure results of inpainted images whose subjective qualities are controlled as the training data. We also propose a masking method for generating training data towards fully automated training data generation. These approaches make it possible to successfully estimate better inpainted images, even though the task is quite subjective. To demonstrate the effectiveness of our approach, we test our algorithm with various datasets and show it outperforms existing IQA methods for inpainting.

3.1 Introduction

The previous chapter proposes the preference order estimation framework aiming at automatically select the best results among several inpainted images with varied parameters. To achieve such automatic selection, we have identified two main issues. The first is that evaluating "correctness" of inpainted results is a task that requires subjective judgment. The second is that even with human judgment, providing absolute scores in a stable manner for inpainted images is a difficult task. Due to these issues, no definitive way has previously been found to estimate subjective quality on the basis of objectively measurable features.

Figure 3.1 explains these two issues with examples. In the figure, Figs. 3.1(c) to (e) show inpainted results obtained with different parameters. These depend on the original image and the original image with the damaged regions respectively

shown in Figs. 3.1(a) and (b). The former issue is explained by the results seen in Figs. 3.1(c) and (d). Although neither of these results are different from the original one shown in Fig. 3.1(a), both of them are perceptually natural. Results such as these are considered to be "correct" as long as they are perceptually natural for humans, even if they differ from the original one [10, 11, 18]. The difficulty in defining this kind of "correctness" means human judgment must be relied on in many cases.



(a) Original image (b) Original image with damaged region (c) Inpainted image 1 (d) Inpainted image 2 (e) Inpainted image 3

Figure 3.1: An example that explains difficulty of evaluating inpainted images objectively.

The latter issue is explained by the results seen in Figs. 3.1(d) and (e). Although differences in quality can be seen between these images, it is quite difficult to stably give absolute scores to them because personal judgment plays such an important role in giving them. Because of these two difficulties, estimating inpainting quality has long been considered a challenging task.

To address the former issue, existing image quality assessment (IQA) methods have tried to find a way to represent subjective quality by means of objectively measurable indicators. Venkatesh and Cheung used observed gaze density inside and outside the damaged region in inpainted images [23]. Instead of observed gaze, many IQA methods using a computational visual saliency map, which simulates human gaze density, have also been proposed [22, 24, 25, 26, 27]. However, actual human gazes vary by viewer and viewing context and their correspondence with saliency maps is quite limited. Isogawa et al. [3] revealed that the pixel-wise unnaturalness that occurs in inpainted images is not suitable for saliency based methods because the resolution of visual saliency maps is coarse. They also proposed perceptually-aware image features focusing on the border area of mask regions where human gazes tend to gather.

To address the latter issue, let us look once more at the Figs. 3.1(d) and (e) results. In this case, although it is not easy to give stable scores, it is comparatively easy to choose which results are better. Current IQA methods mainly focus on providing absolute scores despite the difficulties involved in doing so [26, 27]. They include support vector regression (SVR) based methods and require absolute scores for learning, which is difficult and tends to become unstable. Unlike these methods, that proposed by Isogawa et al. [3] involves an ordering approach, which estimates the preference order of inpainted images. Therefore, to use the method for learning purposes it is only necessary to ascertain the preference order, which is a comparatively easy task. The method uses a learning-to-rank approach and accurately estimates the subjective quality of inpainted images by dividing problems

into a set of pairwise preference order estimation tasks. Learning based methods such as these commonly require a subjective annotation step before the training step, which is considered essential. This labor-intensive annotation leads to both huge annotation cost and, what is worse, fluctuation of evaluation criteria.

Consequently, this chapter proposes a new framework for estimating learning to rank based preference order with automatically generated training data. Such data is referred to as "auto-generated" data in the chapter. The method simulates "failed" inpainting and assumes that a simulated sample has worse subjective quality than the method's best inpainted image. Thus, it generates training pairs automatically without any user intervention.

Contribution: The main contribution we show in this chapter is that the proposed method achieves learning based preference estimation of inpainted images without annotated training data. To the best of our knowledge, this is the first study that tackles learning based estimation of subjective attributes without manually annotated training data. The other contributions include a way to generate degraded inpainted results and a way to generate masked regions as means to fully generate automated training data.

The rest of this chapter is organized as follows. In Section 3.2 we review related work. Section 3.3 describes the learning based ranking method we propose, which is trained with auto-generated data. Section 3.4 proposes a way to generate masked regions with the aim of generating fully automated training data. Section 3.5 verifies the effectiveness of the proposed method and Section 3.6 concludes the chapter with a summary of key points and describes the subjects for future work.

3.2 Related Work

This section introduces related work. First, 3.2.1 overviews existing IQA methods for image inpainting. In 3.2.2 we focus on machine learning, especially as a means to prepare training data. Studies on automatic generation or augmentation of training data are introduced.

3.2.1 Learning Based IQA Methods for Inpainted Image

With the aim of selecting the best one from a plurality of results among varied inpainted images, many IQA methods have been proposed [22, 24, 25, 26, 27, 3]. Among these methods, learning based approaches have demonstrated effective performance [26, 27, 3]. Frantc et al. [26] and Voronin et al. [27] proposed SVR based IQA methods. These approaches estimate an absolute subjective score for each test image. For training regression models, subjectively annotated rating scores are essential. Thus, they used data annotated by subjects who were asked to provide scores on a 5-point scale.

Another learning based approach to tackle this problem is the learning-to-rank approach. It learns and estimates rank order on the basis of a trained ranking function. The important advantage of this approach is that it can learn only on the basis of rank order. Because of this advantage, this approach has been the focus of considerable attention, especially when it is applied to tasks where it is difficult to estimate subjective preference objectively [28, 29, 30, 31]. Isogawa et al. [3] proposed a learning-to-rank based IQA method by pairwise preference estimation, which is written in the previous chapter of this thesis. This method focuses on the premise that the preference order, rather than absolute scores, is good for selecting the best one from a plurality of results, which is the method's primary goal. For training data, the method requires image pairs with annotated preference order.

As described above, one difficulty that commonly exists in learning based methods is the need for a labor-intensive annotation step for obtaining training data. These manual annotations require huge annotation cost. In addition, the judgment criteria of subjects fluctuate occasionally. To overcome these problems, the proposed method enables automatic generation of training data. It generates pairwise training data automatically and applies a learning-to-rank based algorithm to the preference order estimation.

3.2.2 Learning with Auto-Generated Training Set

Larger amounts of training data generally lead to higher performance in learning-based methods. Therefore, in recent years some studies have improved learning accuracy by augmenting the learning data with automatic generation [37, 38]. Pishchulin et al. [37] proposed a human detection and pose estimation by using automatically generated training sets. The main advantage of their method is that it enables human poses and shapes to be controlled explicitly on the basis of existing training sets. They also combine various background images to increase training data. Ros et al. [38] proposed learning based pixel-wise semantic segmentation that uses automatically generated training data. Since annotation data for this task must be provided on a pixel-by-pixel basis, having humans provide the data is labor-intensive. To overcome the problem, they use realistic synthetic images of urban views in a virtual world that can provide annotation data on a pixel-by-pixel basis.

In these tasks, it is apparent that an image generation model can be obtained from annotation data. Thus, generation of training examples, i.e., a set of annotated data and the generated images, is rather easy. In contrast, modeling the relationship between inpainted images and their annotated subjective quality is quite difficult. The reason is shown in Fig. 3.1, where both inpainted images have subjectively good quality. This makes it difficult to create the auto-generated training data. To the best of our knowledge, this study is the first trial of making auto-generated training data for subjectively assessing inpainted image quality.

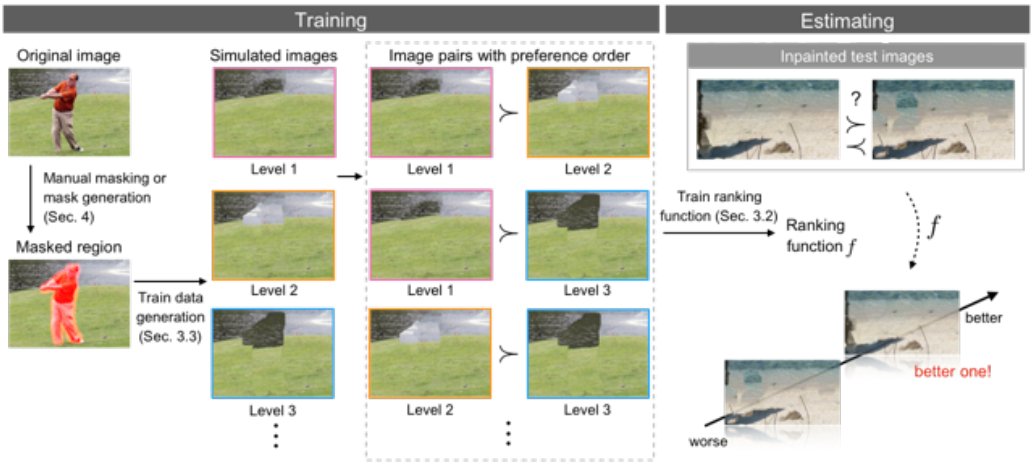


Figure 3.2: Overview of our proposed learning framework

3.3 Proposed Method

3.3.1 Overview

Figure 3.2 shows the overview of the proposed preference ordering framework with automatic generation of training data. Our framework consists of training and estimation phases.

In the training phase, the proposed method first generates a simulated training set. Then, a ranking model is trained with these auto-generated images. As the ranking model, we utilize the pairwise learning-to-rank based method previously used by Isogawa et al. [3]. This is because unlike SVR-based methods [26, 27], a pairwise method only requires a set of preference orders between two images and does not require absolute scores of subjective quality. The learning process is detailed in 3.3.2. Because the proposed method uses pairwise learning-to-rank, the proposed generation of training data yields inpainted image pairs with known preference orders. This is described in more detail in 3.3.3.

The estimation phase procedure is fairly clear. With inpainted image pair input, the method extracts feature vectors that focus on the unnaturalness produced by color or structural discontinuity around the inpainted region contours. Then, the ranking function's scalar output values are calculated. The magnitude relationship between pairs of images shows their preference order.

3.3.2 Model Learning with Auto-Generated Training Data

This subsection describes a learning-to-rank based learning algorithm for ordering pairwise images. The method is trained with auto-generated simulation images that represent degraded inpainted results.

This algorithm premises a ranking function $f(x^i)$ that projects feature vector x^i, which is obtained by $x^i = g(I^i)$ from image I^i, to a one-dimensional axis in accordance with the subjective quality of inpainted results, where $g(\cdot)$ is a feature extraction function. For simplicity, we use "$I^i \succ I^j$" to express that "I^i is preferred to I^j".

For easy understanding, before we describe the ranking algorithm trained with auto-generated training data, let us briefly explain the training algorithm for $f(x)$. We define the function $h(x^i, x^j)$ that denotes preference order as follows.

$$h(x^i, x^j) = \begin{cases} +1 \ (I^i \succ I^j) \\ \ \ 0 \ (\text{no preferences}) \\ -1 \ (I^j \succ I^i), \end{cases} \tag{3.1}$$

The $f(x)$ is trained so that the difference of outputs $f(x^i) - f(x^j)$ has the same sign as $h(x^i, x^j)$. In a word, the function f should satisfy the following formula:

$$\text{sign}(h(x^i, x^j)) = \text{sign}\,(f(x^i) - f(x^j)). \tag{3.2}$$

The goal is to learn f, which is concordant with the training samples. We modeled f with the linear function $f(x) = \omega^\top x$. Then Eq. 4.2 can be rewritten as

$$\text{sign}\,(h(x^i, x^j)) = \text{sign}\,(\omega^\top (x^i - x^j)). \tag{3.3}$$

The error function is defined on the basis of Eq. 4.3 and is optimized with respect to ω. This is the same problem as that of binary classification. We use a pairwise learning-to-rank algorithm called RankingSVM [43] to solve it.

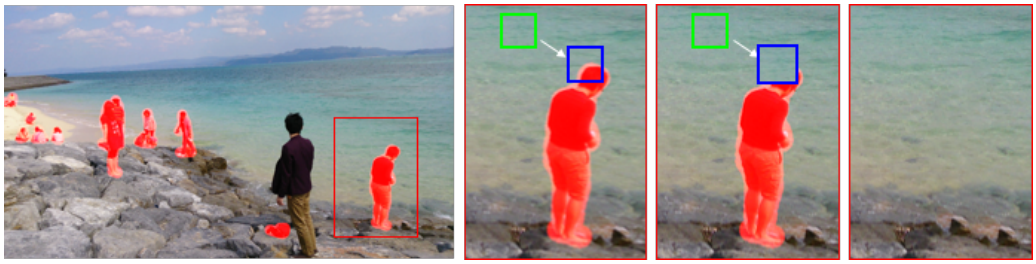
Now we are ready to introduce auto-generated training set into the ranking algorithm. Let I^l_{sim} be a simulated inpainted image with l degraded level, with which larger l indicates more degradation. In accordance with degraded level, the preference order among such images is

$$l_1 < l_2 \rightarrow I^{l_1}_{sim} \succ I^{l_2}_{sim}. \tag{3.4}$$

Such auto-generated images are used to train the ranking function. The way to generate degraded images I_{sim} is described in the next subsection.

3.3.3 Automatic Training Data Generation

The proposed method relies on existing inpainting methods and devices for them to obtain degraded inpainted images that well simulate inpainting failures. Subsection 3.3.3.1 briefly reviews typical inpainting algorithms and then 3.3.3.2 describes how degraded data are generated on the basis of the existing inpainting algorithms.



(a) (b) (c) (d)

Figure 3.3: Typical patch-based approach for image inpainting

3.3.3.1 Patch Based Image Inpainting Algorithms

Among various inpainting methods, patch-based algorithms are widely acknowledged as promising approaches. Typically they comprise three steps, which we will explain by using Fig. 3.3. For the damaged region masked with red in Fig. 3.3(a), (1) a patch that includes both the source and the damaged region is set as a target, the blue rectangle in Fig. 3.3(b), (2) a similar patch for the target patch, the green rectangle in Fig. 3.3(b), is retrieved in the source region, and (3) the damaged region in the target patch is replaced in accordance with similar patches as in Fig. 3.3(c). The resultant restored image is shown in Fig. 3.3(d).

For proposing our method in subsequent part, we will introduce the following notation for the above patch-based inpainting concept. Here, $P(p)$ is a target patch whose center pixel is p, and $\widehat{P}_p^N$ denotes the N-th most similar patch from $P(p)$. Since total inpainting quality highly depends on the quality of retrieved patches, it is basically acknowledged that the more similar the retrieved patch is, the better the inpainted quality becomes. That is, for "fine" inpainted results, the most similar patch (i.e., with $N = 1$) from $P(p)$, which is denoted as $\widehat{P}_p^1$, is used as in Eq. 3.5 to restore a missing region.

$$\widehat{P}_p^1 = P(q') = \underset{P(q)}{\operatorname{argmin}}\ dist(P(p), P(q)) \tag{3.5}$$

$dist(\cdot)$ represents distance function. The proposed method uses the assumption in an inverse way, i.e., dissimilar patches generate unnatural inpainted images.

3.3.3.2 Auto-Generated Inpainted Images as a Training Set

In simulating failed inpainted images, we found that if we selected the N-th most similar patch $\widehat{P}^N$ having larger N, it would apparently correspond to the cases in which good patches for inpainting cannot be found. This is a typical case of inpainting failure. Therefore, as the value N gets larger, the patches become dissimilar and the inpainting results get worse. That is, simulated inpainted images are generated so that their relationships depend on the level of patch similarity as $I_{sim}^{l_1} \succ I_{sim}^{l_2}$ when $l_1 < l_2$, where I_{sim}^N represents a simulated image inpainted with

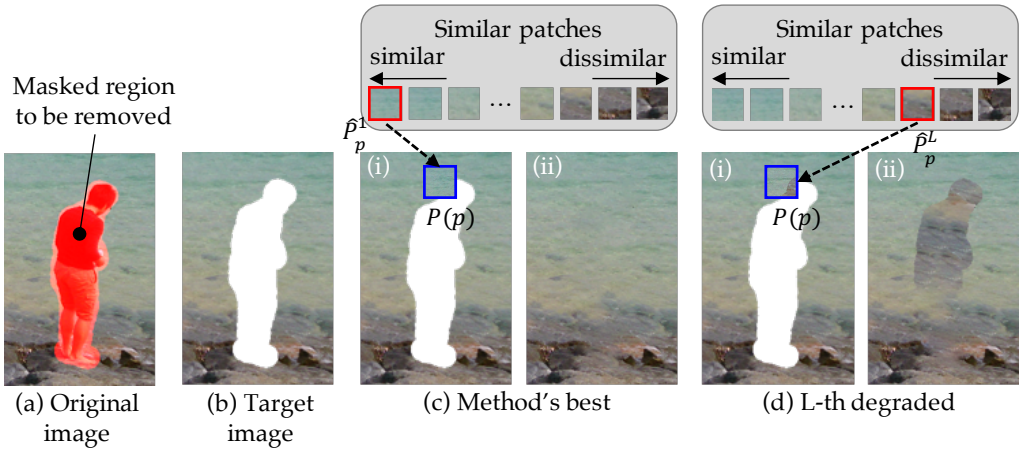


Figure 3.4: Patch based degraded inpainted image generation.

N-th similar patches. We propose incorporating this patch retrieval into existing inpainting algorithms. The following shows our simulated image generation with two types of algorithms as examples.

With Patch-Retrieval Based Method. Here we explain our data simulation for patch retrieval based inpainting methods with Criminisi et al's method [6] as a base algorithm. This simulation can also be applied to other patch-based algorithms. The original method uses simple patch retrieval as shown in Eq. 3.5. Our method can be easily incorporated into this patch retrieval; instead of retrieving the most similar patches $\widehat{P}_p^1$, we obtain $\widehat{P}_p^L$ with $L > 1$. Figure 3.4 illustrates this in more detail. The original image with damaged region is shown in (a). The resultant inpainted images are shown in (b) and (c). Here, (b) is the method's best result and (c) is a simulated deteriorated result obtained using our proposed method. As the typical procedure of patch-based inpainting, target patch $P(p)$ is determined in (i), and then a similar patch is retrieved and used for filling in the hole. In case (b), the most similar patch is used. Unlike this, the proposed method uses a dissimilar patch depending on degraded level L and obtains the degraded result as shown in (c-ii).

With Image-Retrieval Based Method. Some current studies extend the basic algorithms by using patch retrieval indirectly. Our training data simulation method can also be applied to such methods without loss of generality. Here we explain an extension using He and Sun's method [10] as the base algorithm. He and Sun improve the basic algorithm on the basis of two ideas; extension of the patch P to the whole image I, and treating an inpainting task as a Photomontage problem [46]. Figure 3.5 illustrates this in more detail. The original image with damaged region and the target region to be inpainted are shown in Figs. 3.5(a) and (b). The resultant inpainted images are shown in Figs. 3.5(c) and (d). Fig. 3.5(c) is the method's best result and Fig. 3.5(d) is a simulated deteriorated result obtained using our proposed method. Let $\widehat{I}^N$ be the N-th most similar image for damaged image I. To generate the method's best inpainting result, they retrieve the K most similar images $\widehat{I}_K = \{\widehat{I}^1, \widehat{I}^2, ..., \widehat{I}^K\}$ and the missing region is filled by combining

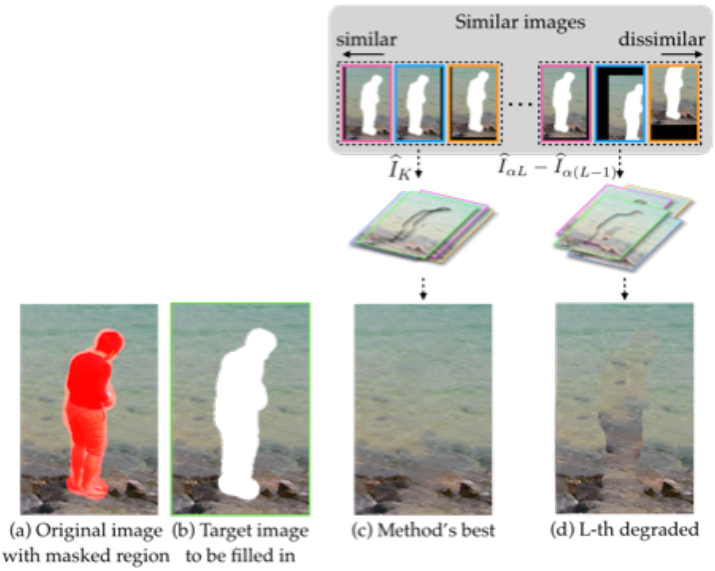


Figure 3.5: Image retrieval based degraded inpainted image generation.

a stack of these images. Our method modifies this image retrieval part; instead of retrieving the K most similar images, we obtain $\widehat{I}_{\alpha L}$ that excludes $\widehat{I}_{\alpha(L-1)}$. That is, we obtain $\widehat{I}_{\alpha L} - \widehat{I}_{\alpha(L-1)} = \{\widehat{I}^{\alpha(L-1)+1}, \widehat{I}^{\alpha(L-1)+2}, ..., \widehat{I}^{\alpha L}\}$ to generate an L-th level of a degraded image.

3.4 Towards Completely Automatic Training Data Generation: Masked Region Generation

The previous section focuses only on the labor cost involved in annotating preference orders. Since training data generation requires huge annotation cost, our approach with auto-generated data can significantly reduce the cost. However, there is still a process that must be done manually in this method. That is to designate the region to be inpainted as a masked region. This operation requires manual intervention and hinders larger training set generation.

To improve our method, we also propose a method to generate masked regions for effective auto-generated data by utilizing semantic segmentation. By additionally using this automatic designation of mask regions, the method eliminates any manual work needed to generate training data. In other words, it makes it possible to increase the amount of training data with no labor cost.

However, it is known that the quality of inpainted results varies largely depending on their masked region. Since our training data generation method assumes that simulated image pairs with multi-levels of degradation have orders in terms of quality, the method's best inpainted images with no degradation should have good enough quality with its masked region. In case the method's best inpainting results with degradation level $L = 0$ do not have good enough quality, the quality

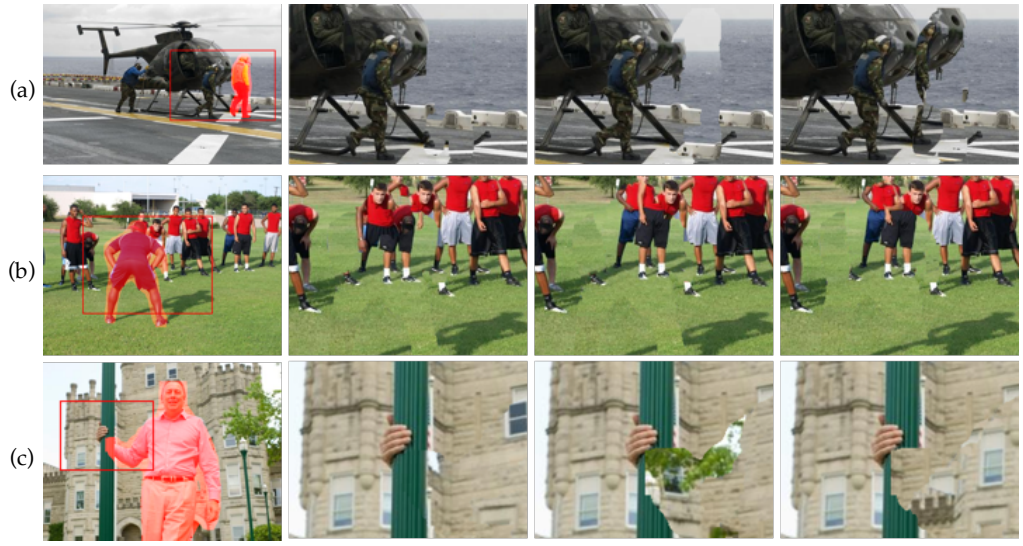


Figure 3.6: Example of auto-generated data quality depending on masked region; (a) desirable multi-degraded auto-generated data, (b)(c) undesirable auto-generated data with masked region that do not satisfy the requirements.

of all degraded image is biased toward poor directions and thus it might break our assumption.

Thus, we consider that desirable masked regions for effective auto generated data should satisfy the following three requirements; (1) the images for training data include varied scenes for high versatility training sets, (2) other objects are not adjacent to the contours of the masked regions, and (3) the region of the object to be inpainted does not protrude from the masked region.

Figure 3.6 explains how requirements (2) and (3) affect auto-generated data. In the figure, (a) shows the masked region that satisfies the requirement and its auto-generated data with degradation level 0 to 2. Multi-levels of degraded images become worse if degradation level L increases, as expected. Figs. 3.6(b) and (c) respectively show failure cases in which requirements (2) or (3) are not satisfied. In Fig. 3.6(b) another object touches a masked human region, and in Fig. 3.6(c) the people's region to be inpainted is revealed from the masked region. In both cases, unnatural inpainted results are generated in all degradation stages and there are no definitive preferences between them. This is due to the fact that the designated masked region makes it difficult to find the source region to be used for filling the hole. Restoring missing objects such as adjacent humans or revealed human regions is rather difficult.

To satisfy the requirements described above, we use people's regions in images found by semantic segmentation. Humans appear in a lot of images and they are less likely to be adjacent to other objects compared to other objects such as desks in a classroom. Therefore, using such regions makes it possible to meet the

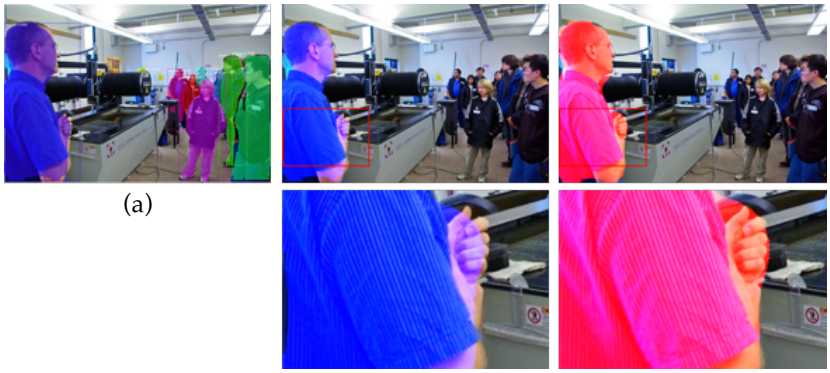


Figure 3.7: Proposed masked region generation. The method first detects people's regions as shown in (a) and dilates initial masked region (b) to (c) so that the region satisfies the mask requirements.

first and second requirements. In order to meet the second point better, we use regions that do not have any adjacency with other objects. To satisfy the third requirement, we dilate the extracted region since the human region extracted by semantic segmentation is often smaller than that of an actual human region.

The detailed process of masked region generation is described below (see Fig. 3.7). With original image I_{orig} obtained from a dataset, the Internet, etc (see Fig. 3.7(a)), semantic segmentation results are calculated as shown in Fig. 3.7(b). We use Mask R-CNN [47] as a semantic segmentation method. If the detected people's region has no overlap with any other object and its size is 1 to 20 % of I_{orig}, the initial masked image I^{init}_{mask} is generated with the segmented region (see Fig. 3.7(c)). For I_{orig} with multiple people's regions, only the region having the largest area is adopted. The final masked image I_{mask} shown in Fig. 3.7(d) is calculated by dilating the masked region in I^{init}_{mask}.

3.5 Experiment

This section reports the efficacy of the proposed method. In Subsection 3.5.1 we will show the experimental setup we used, including training data preparation. In 3.5.2 we will show the efficacy of our auto-generated data as a training set. In 3.5.3 we will demonstrate the efficacy of the proposed masked region generation.

3.5.1 Experimental Setup

3.5.1.1 Ranking Learning

The proposed method uses Isogawa et al.'s rank learning framework [3]. The characteristics and advantage of the framework are as follows. The framework

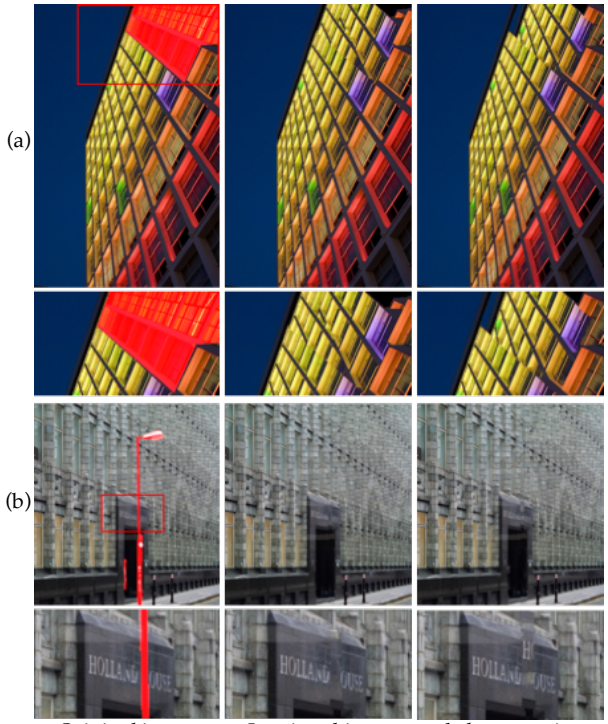


Figure 3.8: Inpainted image pairs, the preference orders of which were correctly estimated with our model even with (a) non-uniformity background textures, and (b) complex shape of masked region.

uses RankingSVM, implemented using SVM Rank [44] with a radial basis function (RBF) kernel, whose parameters are well tuned. For training and testing, it uses ten-dimensional image features that focus on the unnaturalness around the contours of the inpainted regions produced by color or structural discontinuity. Since the features are normalized for the size of the masked region contour, the rank learning is relatively robust for the shape complexity of the masked region or non-uniformity of the texture (see Fig. 3.8 for image pairs, the preference orders of which were correctly estimated).

3.5.1.2 Preparing Manually Annotated Data

Manually annotated data were basically used for test data with ground-truth preference annotation. They were also used as training data for comparing the estimation accuracy obtained with an auto-generation data based model to that obtained with a manual annotation data based one.

Subjective Annotation. We prepared 100 publicly available images obtained from the Web. Damaged regions in these images were manually masked. For

each masked image, we generated a fixed number of inpainted results with different parameters. The number differed depending on the experiments we conducted. The experiments discussed in 3.5.2.1 and 3.5.2.2 and the first experiment discussed in 3.5.2.3 involved generating six inpainted results by a combination of three options of patch size and two options of number of similar images to be retrieved. The second experiment discussed in 3.5.2.3 involved preparing three inpainted results by changing the pre-trained model for inpainting. The third experiment discussed in 3.5.2.3 also involved preparing three inpainted results by changing the inpainting method.

The quality of the images was evaluated by eight subjects (four males and four females) with normal vision. To make the users' judgment easy, we randomly displayed a pair of inpainted images side-by-side. Subjects were asked to choose one of three options: right image is better, left image is better, and no preference order (i.e., it is hard to decide which one is better or which one is worse). As inpainting methods, we basically used He and Sun's method [10] throughout this section, but in 3.5.2.3 we add Huang et al.'s method [11], Herling et al.'s method [12], and Yu et al.'s method [14].

Notation and Reliability of Annotated Data. We used $T_a^{(M,S)}$ to denote annotated datasets with inpainted method M, for which a consensus was obtained for at least S subjects. For example, $T_a^{(He,8)}$ indicates the dataset with He et al's inpainting method, which got a unanimous answer from all eight subjects. With the dataset $T_a^{(He,5)}$ consensus was obtained for from five to eight subjects. Thus, more than half of the subjects gave the same preference order to pairs in $T_a^{(He,5)}$. This indicates that S reflects the difficulty humans have in making judgments in such cases.

3.5.1.3 Auto-Generated Training Data

This subsection describes how we got an auto-generated training set.

Degraded Image Simulation. We gathered the same images of annotated data. But, please note that we excluded auto-generated data simulated with identical images as test data. Damaged regions in these images were manually masked. Since the position, size, and shape of the damaged region are normalized during the learning-to-rank process, we were able to set the damaged regions arbitrarily regardless of the objects in the target images. We set five degradation levels for simulating inpainted images I_{sim}^L, i.e., $L = 0, 1, 2, 3$, and 4, where $L = 0$ indicates an image without any intentional degradation, i.e., the method's best inpainted image. These five image levels are generated from one original image. By combining these five images, we generated ${}_5C_2 = 10$ pairs of training data, with preference orders $I_{sim}^x \succ I_{sim}^y$ $(\forall x < y)$, i.e., the inverse of degradation level.

Figure 3.9(b) shows degraded images depending on the degraded level L ($L = 0, 1, 2, 3$, and 4). All degraded images are inpainted with the masked region

Figure 3.9: Simulated inpainted images.

Table 3.1: Prediction accuracy comparison with existing image quality assessment metrics. The highest scores are underlined. [%]

	$T_a^{(He,5)}$	$T_a^{(He,6)}$	$T_a^{(He,7)}$	$T_a^{(He,8)}$
ASVS [22]	45.11	44.85	43.66	44.74
DN [22]	53.26	53.68	56.34	57.89
$\overline{GD}_{in}$ [23]	43.48	44.85	40.85	39.47
BorSal [24]	42.39	43.38	42.25	44.74
StructBorSal [24]	46.74	45.59	42.25	52.63
RankIQA [48]	65.79	60.53	63.16	42.11
Isogawa et al. [3]	60.33	62.5	71.83	76.32
Ours(T_d^{He})	<u>66.85</u>	<u>70.59</u>	<u>76.06</u>	<u>78.95</u>
Ours($T_d^{He} + T_a^{He}$)	65.22	68.38	<u>76.06</u>	<u>78.95</u>

shown in (a). Figure 3.9 shows that our method simulates degraded images well; each degraded image gets worse quality as L increases. Though the deterioration is subjective, it well simulates the failures that typically occur in ordinary inpainting methods having inappropriate parameters such as patch size.

Notation. We denote the auto-generated degraded images with inpainting method M for training T_d^M. For example, auto-generated data with He et al's inpainting method is denoted as T_d^{He}.

3.5.2 Investigation to Ascertain Effectiveness of Auto-Generated Training Data

3.5.2.1 Comparison with Existing IQA Methods

We conducted experiments comparing our method to other IQA methods for image inpainting, i.e., *ASVS* and *DN* by Ardis and Singhal. [22], $\overline{GD}_{in}$ by Venkatesh and Cheung [23], *BorSal*, *StructBorSal* by Oncu et al. [24] as non learning-based

methods, and Isogawa et al.'s method [3] as a learning-based method. We also verified RankIQA [48], the rank-learning-based IQA method with a deep neural network (DNN). Although the method is not for IQA of inpainting, we argue that the comparison with the DNN-based IQA method is informative. Note that although the $\overline{GD}_{in}$ originally uses measured human gaze, we used a saliency map instead. This is the same evaluation approach used in Oncu et al. [24]. For training with Isogawa et al.'s method we used the annotation data of $T_a^{(He,8)}$. Our proposed learning method trained with auto-generated data T_d^{He} is denoted as $Ours(T_d^{He})$.

Table 3.1 shows the prediction accuracy for all test data $T_a^{(He,S)}$ $(5 \leq S \leq 8)$ obtained for each metric. Excluding inpainted images with extremely poor quality, the amounts of test data $|T_a^S|$ of T_a^S with $S = 5$ to 8 were $(|T_a^5|, |T_a^6|, |T_a^7|, |T_a^8|) = (184, 136, 71, 38)$. Our method $Ours(T_d^{He})$ correctly estimated the preference order within image pair with the highest score for all test data; the improvement our method achieved over Isogawa et al.'s method was 6.52, 8.09, 4.23, and 2.63 points for test data $T_a^{(He,5)}$, $T_a^{(He,6)}$, $T_a^{(He,7)}$, and $T_a^{(He,8)}$.

Figure 3.8 shows examples of image pairs, the preference orders of which were correctly estimated even with non-uniformity texture (see (a)) and shape complexity of masked region (see (b)). Please refer our supplemental material for more results.

3.5.2.2 Verifying Effectiveness of Auto-Generated Training Data Depending on Varied Conditions

This subsection describes three more investigations we conducted to verify the validity of the auto-generated data. The first one investigates the effects of the volume of auto-generated data on estimation accuracy, which is the ratio of estimation success of preference order among annotated pairs. Figure 3.10 shows the estimation accuracy obtained when the amount of auto-generated data is increased from 50 to 990 in 50 increments. Training data are randomly selected from T_d^{He}. As shown in the graphs in the figure, estimation accuracy increased as the amount of training data increased. In addition, to investigate whether auto-generated data can be used as a substitute for annotated data, the performances depending on the proportion of auto-generated data were tested. Figure 3.11 shows performances depending on the proportion of auto-generated data with $T_a^{(He,5)}$. Here, the number of training data was fixed to 990 in all cases; only the proportion of annotated and auto-generated data was changed. The amount of annotated data was decreased from 180 to 0 in 10 decrements. Even though the amount of subjectively annotated training data was changed, the estimation accuracies were almost constant in all cases. These results suggested that auto-generated training data could be a substitute for manually annotated data.

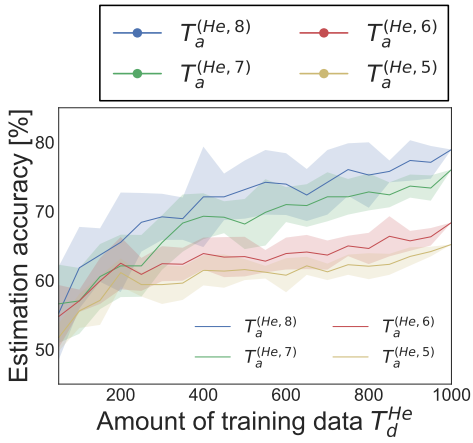


Figure 3.10: Prediction accuracy with each T_a^{He} depending on the amount of T_d^{He}.

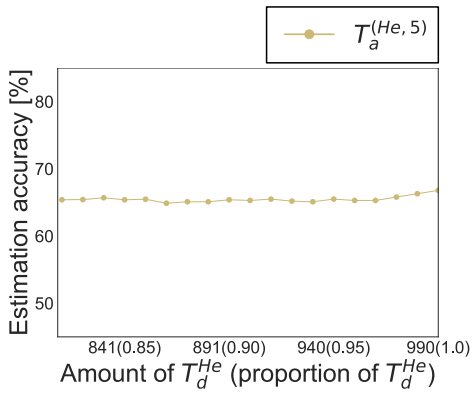


Figure 3.11: Prediction accuracy depending on the proportion of T_d^{He}.

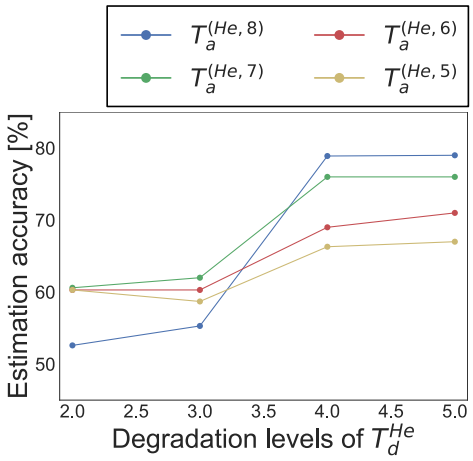


Figure 3.12: Prediction accuracy depending on the levels of T_d^{He}.

We also investigated how the amount of auto-generated data affected estimation accuracy. Figure 3.12 shows estimation accuracy for each T_a^S when the auto-generated training data levels L were changed between $L = 1$ to 4. The available training data amounts were respectively 99, 297, 594, and 990 for $L = 1$, 2, 3, and 4. However, to focus on the affect of data levels, we set it to the smallest number 99, i.e., that for $L = 1$. Note that the results are average performances with 10 trial runs and that training data for each trial are randomly selected. As the figure shows, the estimation accuracy increases as L increases, which suggests that multi-levels of auto-generated data work effectively. These two kinds of investigations suggest that the auto-generated data works as expected and using multi-levels of data works effectively.

The third one verifies the performance when subjectively annotated data is added to auto-generated data. We added $T_a^{(He,S)}$ to T_d^{He} for further verification of auto-generated data performance. Hereafter, in this section we denote these two data as T_a^S and T_d for simplify the explanation. We denote our learning method with such

Table 3.2: Prediction accuracy with or without subjectively annotated data. The highest scores are underlined. [%]

	T_a^5	T_a^6	T_a^7	T_a^8
Ours(T_d)	<u>66.85</u>	<u>70.59</u>	<u>76.06</u>	<u>78.95</u>
Ours($T_d + T_a$)	65.22	68.38	<u>76.06</u>	<u>78.95</u>

Table 3.3: Prediction accuracy with or without unreliable data. The highest scores are underlined. [%]

	T_a^5	T_a^6	T_a^7	T_a^8
Ours(T_d^{re})	<u>66.30</u>	<u>69.85</u>	<u>76.06</u>	<u>81.58</u>
Ours(T_d^{un})	63.04	67.65	73.24	78.95

data as $Ours(T_d + T_a)$. The comparison between $Ours(T_d)$ and $Ours(T_d + T_a)$ with all test data T_a^S with $S = 5, 6, 7$, and 8 is shown in Table 3.2.

With this table, we found that the use of annotated training data does not show significant changes on prediction accuracies of for all cases (T_a^5, T_a^6, T_a^7, and T_a^8). However, in case of low consensus data such as $S = 5$ and 6, the use of annotated training data deteriorated the prediction accuracy. "Low consensus" means the subjective judgement varies by the subject and may not suit for machine learning. To verify this consideration, we conducted the next experiment.

We divide the auto-generated data, T_d into two groups; reliable data set consisted by auto-generated data of L =1 and 4, T_d^{re}; unreliable data set consisted by that of L =2 and 3, T_d^{un}. We subjectively confirmed that T_d^{un} have small difference in subjective quality and are difficult to be judged by a large margin compare to T_d^{re}, as in Fig. 3.9. The prediction accuracy is shown in Table 3.3. For all test data, $Ours(T_d^{re})$ excels $Ours(T_d^{un})$. It also suggests that subjectively similar data like T_d^{un} is not a good data for training. Thus, we should consider the balance between the number of auto-generated data and the quality of them as a future work.

3.5.2.3 Effectiveness for Test Image with Unknown Inpainting Method

Up to the preceding sections, we have examined the effect of auto-generated training data under the condition that the inpainting method used for test and training data generation is same. However, since inpainted results vary depending on their inpainting method, auto-generated training data are also varied by their inpainting methods. Thus, this section investigates how ranking function trained with a certain method works for the test data generated by another method. Hereafter, we define "pre-trained" and "re-trained" ranking functions as that are trained without/with auto-generated training data inpainted with same method as test data. We also denote test data whose inpainting method is not used for pre-training

Table 3.4: Prediction accuracy with or without auto-generated data inpainted with unknown Huang et al's method [%]

	$T_a^{(Sc,5)}$	$T_a^{(Sc,6)}$	$T_a^{(Sc,7)}$	$T_a^{(Sc,8)}$
Ours(T_d^{He})	71.03	74.37	81.48	82.98
Ours(T_d^{He+Sc})	69.16	71.86	77.78	82.98

Table 3.5: Prediction accuracy with or without auto-generated data inpainted with unknown Herling et al's method [%]

	$T_a^{(Pix,5)}$	$T_a^{(Pix,6)}$	$T_a^{(Pix,7)}$	$T_a^{(Pix,8)}$
Ours(T_d^{He})	66.41	67.61	70.28	72.87
Ours(T_d^{He+Pix})	62.92	63.46	68.11	71.29

as "unknown test data" or "test data with unknown method", and its inpainting method as "unknown inpainting method".

With Patch-Based Inpainting Methods. For investigation, we used Huang et al.'s [11] and Herling et al's [12] methods as unknown patch-based inpainting methods. Since these two are patch-retrieval based methods, our patch-retrieval based simulation method is applied for auto-generated data preparation. Such auto-generated data with these two methods were not used for pre-training and we used He et al.'s method [10] for pre-training same as previous section. From here we denote auto-generated training data with these three inpainting methods as T_d^{He}, T_d^{Sc}, and T_d^{Pix}. Samely, we denote test data with new two methods as T_a^{Sc} and T_a^{Pix}. As a ground truth, preference orders for T_a^{Sc} and T_a^{Pix} are annotated by 8 subjects. T_a^{Sc}, T_a^{Pix} with at least S subject's consensus is denoted as $T_a^{(Sc,S)}$ and $T_a^{(Pix,S)}$. The amounts of test data of $T_a^{(Sc,S)}$ and $T_a^{(Pix,S)}$ with $S = 5$ to 8 were (321, 199, 108, 47) and (712, 602, 461, 317), respectively. Same as previous section, 100 original images for auto-generated training data were prepared for five levels of degraded images. Each of T_d^{He}, T_d^{Sc}, T_d^{Pix} includes ${}_5C_2 \times 100 = 1000$ pairs of training data.

Test data with unknown inpainted method T_a^{Sc}, T_a^{Pix} are evaluated under two conditions; one is with pre-trained ranking model with T_d^{He}, and the other is re-trained model with same inpainting method as test data, i.e., mixture training data includes T_d^{He} and T_d^{Sc}, or T_d^{He} and T_d^{Pix}. We call such adjacent data as "mixture training data" and denote them as T_d^{He+Sc} and T_d^{He+Pix}, respectively. These mixture training data have twice samples of each training data, i.e., 2000 pairs.

Figures 3.13 and 3.14 show estimation accuracy for each T_a^S where $S = 5, 6, 7,$ and 8 with pre-trained ranking function with T_d^{He} (see (a)) and re-trained it with mixture training data T_d^{He+Sc} or T_d^{He+Pix} (see (b)). In the figures, vertical axis shows estimation accuracy and horizontal axis shows amount of training data, which is auto-generated. That are increased from 100 to 1000 in 100 increments

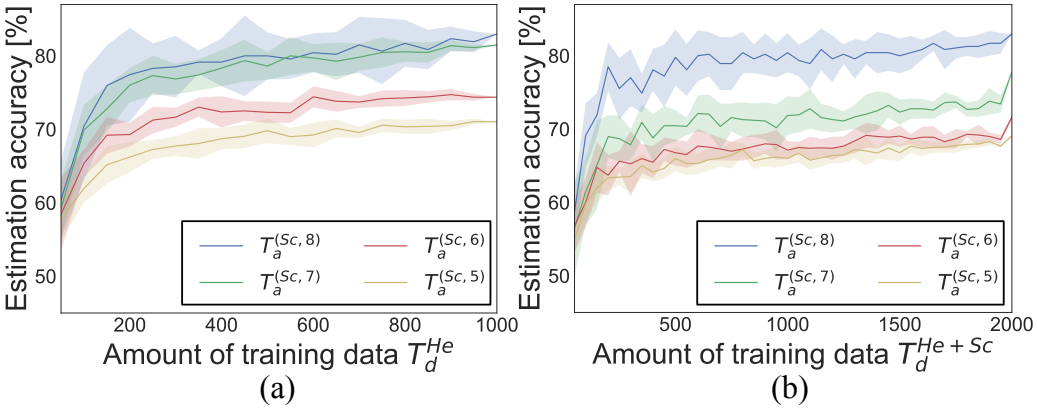


Figure 3.13: Prediction accuracy for T_a^{Sc} with (a) pre-trained model with T_d^{He} and (b) re-trained model with T_d^{He+Sc}.

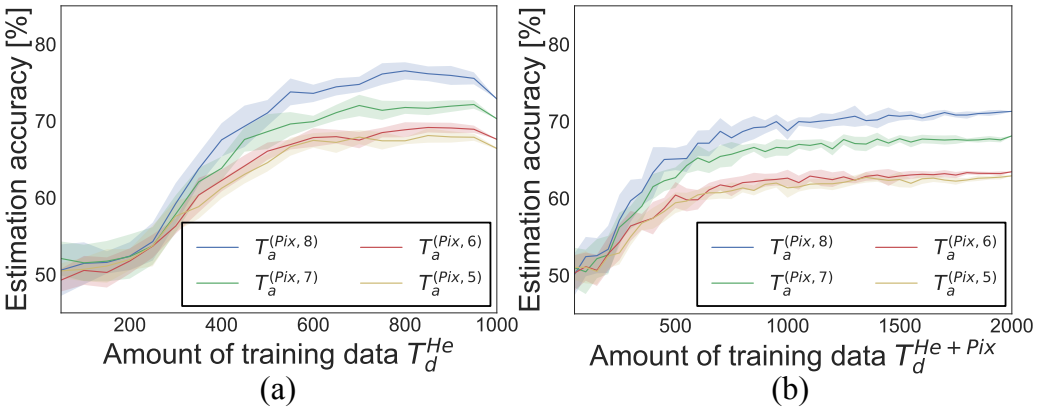


Figure 3.14: Prediction accuracy for T_a^{Pix} with (a) pre-trained model with T_d^{He} and (b) re-trained model with T_d^{He+Pix}.

for (a) and from 100 to 2000 in 100 increments for (b). The training data are randomly selected and the plotted accuracy is an average of 10 runs with these standard deviations. Tables 3.4 and 3.5 show the estimation accuracy value we obtained for each T_a^S, where $S = 5, 6, 7$, and 8 with the pre-trained ranking function with T_d^{He} with 1000 data elements and re-trained it with mixture training set T_d^{He+Sc} or T_d^{He+Pix} with 2000 data.

As the results show, it was possible to estimate preference orders with high accuracy, i.e., 82.97% for T_a^{Sc} and 72.87% for T_a^{Pix} even for these unknown test data. In addition, regarding result with T_d^{He+Sc}, although estimation accuracy was slightly decreased with mixture training data, significant differences could not be observed. These results show that our method has a certain amount of robustness against the patch-based inpainting method used.

With GAN-Based Inpainting Methods. So far, we have used specific types of inpainting methods, i.e., image retrieval based and patch based. However,

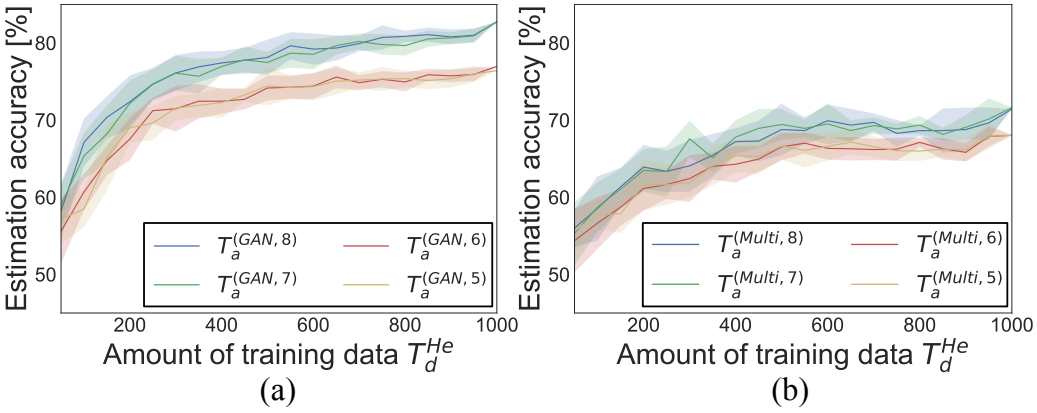


Figure 3.15: Prediction accuracy for (a) T_a^{GAN} and (b) T_a^{Multi}. T_d^{He} was used as trained model

in addition to these effective conventional methods, generative adversarial network (GAN) based inpainting methods have achieved remarkable progress in recent years. We verified how our IQA method trained with auto-generated data works for GAN-based inpainting methods that may have different types of degradation.

We used Yu et al.'s [14] GAN-based inpainting method as the unknown method. The method was trained with the three database, i.e., places2 [49], CelebA [50], and imageNet [51]. As in Sections 3.5.2.1 and 3.5.2.2, we used He et al.'s method [10] for training our preference-ordering model. The denotation for auto-generated data is the same as previous one; T_d^{He}. Test data with Yu et al.'s GAN-based inpainting method are denoted as T_a^{GAN}, and T_a^{GAN} with the consensus of at least S participants are denoted as $T_a^{(GAN,S)}$. The amount of $T_a^{(GAN,S)}$ with $S = 5$ to 8 was (212, 204, 151, 139). As discussed in the previous section, 100 original images for auto-generated training data were prepared for five levels of degraded images; thus, T_d^{He} includes ${}_5C_2 \times 100 = 1000$ pairs of training data.

Figure 3.15(a) shows the estimation accuracy for each $T_a^{(GAN,S)}$, where $S = 5, 6, 7$, and 8 with the pre-trained ranking function with T_d^{He}. The vertical axis shows estimation accuracy and the horizontal axis shows the amount of training data, which were auto-generated. The amount of training data were increased from 100 to 1000 in 100 increments. The training data were randomly selected, and the plotted accuracy is an average of 10 runs with these standard deviations. Table 3.6 shows the estimation accuracy score for each $T_a^{(GAN,S)}$, where $S = 5, 6, 7$, and 8 with the pre-trained ranking function with T_d^{He}. Even with the GAN-based inpainted images, it was possible to estimate preference orders with high accuracy, i.e., 82.73%. The results also indicate that our method can handle test images with different types of inpainting methods with a certain amount of robustness.

With Multiple Types of Inpainting Methods for Test Images. So far, we have focused on one inpainting algorithm for one experiment to generate test images. That is, our ranking algorithm estimates preference orders of images inpainted

Table 3.6: Prediction accuracy for test sets generated with unknown inpainted methods, i.e., Yu et al.'s GAN-based inpainting method [%]

	$T_a^{(GAN,5)}$	$T_a^{(GAN,6)}$	$T_a^{(GAN,7)}$	$T_a^{(GAN,8)}$
Ours(T_d^{He})	76.42	76.96	82.78	82.73

Table 3.7: Prediction accuracy for test sets generated with multiple unknown inpainted methods, i.e., Yu et al.'s GAN-based, He et al.'s image retrieval based, and Huang et al's patch based methods[%]

	$T_a^{(Multi,5)}$	$T_a^{(Multi,6)}$	$T_a^{(Multi,7)}$	$T_a^{(Multi,8)}$
Ours(T_d^{He})	68.02	68.04	71.63	71.43

with the same algorithms. However, there may be situations in which users want to find the orders between images inpainted **with different algorithms.** To generate test images, this experiment uses three different types of inpainted algorithms, i.e., patch based [11], image retrieval based [10], and GAN based [14], trained with the places2 dataset [49]. As in the previous experiments, we used He et al.'s method [10] for auto-generated training data. The auto-generated training and test sets consisting of images with the three different inpainting methods are respectively denoted as T_d^{He} and T_a^{Multi}. The T_a^{Multi} with the consensus of at least S participants is denoted as $T_a^{(Multi,S)}$. The amount of test data of $T_a^{(Multi,S)}$ with $S = 5$ to 8 was (197, 194, 141, 140). Training data consisted of ${}_5C_2 \times 100 = 1000$ pairs, as in the previous experiments.

Figure 3.15(b) shows the estimation accuracy for each $T_a^{(Multi,S)}$ where $S = 5, 6, 7,$ and 8. The vertical axis shows estimation accuracy and the horizontal axis shows the amount of training data, which were auto-generated. That are increased from 100 to 1000 in 100 increments. The training data were randomly selected, and the plotted accuracy is an average of 10 runs with these standard deviations. Table 3.7 shows the estimation accuracy score for each $T_a^{(Multi,S)}$, where $S = 5, 6, 7,$ and 8. Despite the task's difficulty in estimating preference orders between images generated using different inpainting methods, our proposed method estimated the orders with 71.43%. Note that the test images contained inpainted results with two unknown and fundamentally different algorithms from that for training-data generation. The results also indicate that our method can work well even for a test data set consisting of images inpainted using different types of methods.

3.5.3 Effectiveness Investigation for Masked Region Generation

This section investigates the effectiveness of proposed masked region creation towards completely human labor-free training data generation. Note that our main

Figure 3.16: Auto-generated training data with proposed masked region generation; (a) generated I^{people}_{mask}, (b) close-up view of multi levels of simulated inpainted images.

proposal is a training-data-generation method with the degraded inpainting introduced thus far. It is not restricted to fully automatic masking, which is introduced in this section.

Experimental Setup. For experiments, we additionally gathered new images from ImageNet [51]. The mask type, i.e., the masked images of the proposed method and comparison targets, are as follows.

1. The proposed method is denoted by I^{people}_{mask}, which uses automatically segmented people regions without contacting other objects (proposed in Sec. 3.4).
2. The first method for comparison is denoted by I^{rect}_{mask}, which is masked by rectangle of 200×200 [pixels] centered at the original image.
3. The second method for comparison is denoted by $I^{people(ad)}_{mask}$, which is similar to the proposed method but includes images that are adjacent to other objects.
4. The third method for comparison is denoted by I^{all}_{mask}, which is also similar to the proposed method. Although the proposed method only used people's region, it uses all kinds of objects that are automatically tagged.

Training Data Generation. Inpainted images of five levels of degradation with use of He et al.'s method were generated for each mask type. These training data that corresponds to I^{people}_{mask}, I^{rect}_{mask}, $I^{people(ad)}_{mask}$, and I^{all}_{mask} are denoted by T^{people}_{d}, T^{rect}_{d}, $T^{people(ad)}_{d}$, and T^{all}_{d} respectively. The amount of original images used for

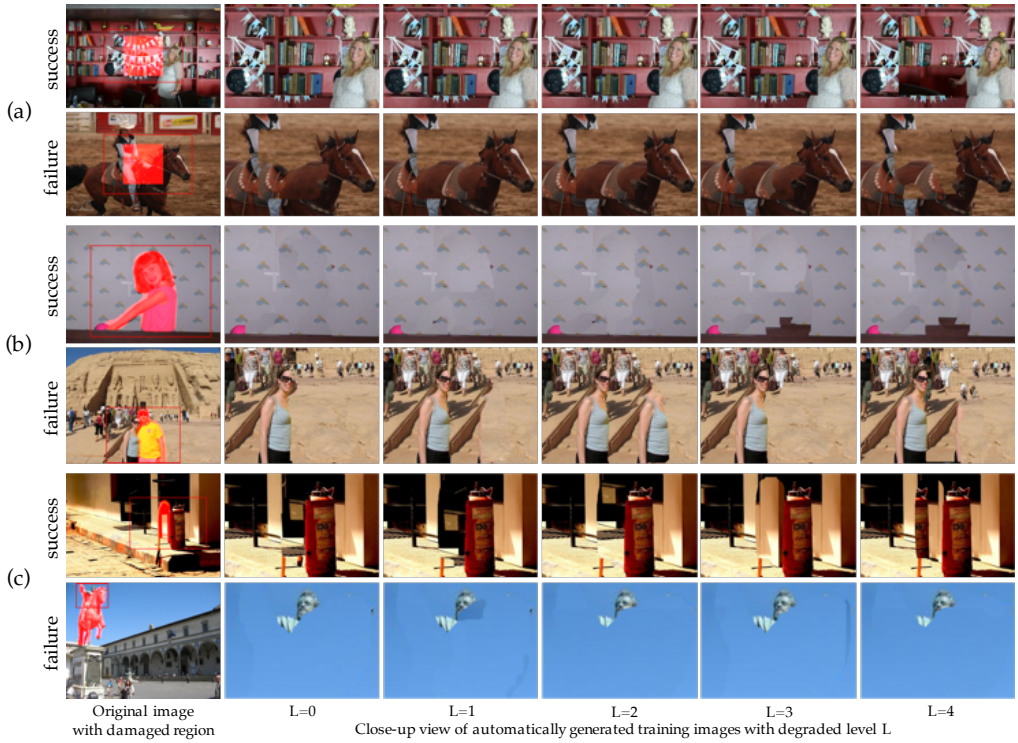


Figure 3.17: Success or failure case of auto-generated training data with automatically generated masked region (a) I_{mask}^{rect}, (b) $I_{mask}^{people(ad)}$, and (a) I_{mask}^{all}.

auto-generated data were 900 for each type of mask and thus each auto-generated training set consisted of ${}_5C_2 \times 900 = 9000$ pairs of training data. Examples of training data T_d^{people} with degraded levels L are shown in Fig. 3.16. As these examples show, degraded inpainting images as training data are appropriately generated. Examples of other training set, i.e., I_{mask}^{rect}, $I_{mask}^{people(ad)}$, and I_{mask}^{all}, are shown in Fig. 3.17.

Results. Figure 3.18 shows estimation accuracy for each T_a^S with He et al.'s method where $S = 5, 6, 7$, and 8 with each dataset T_d^{people}, T_d^{rect}, $T_d^{people(ad)}$, and T_d^{all}. In all of these graphs, vertical axis shows estimation accuracy and horizontal axis shows amount of data used for training, that are increased from 100 to 9000 in 100 increments. The training data are randomly selected and the plotted

Table 3.8: Prediction accuracy comparison for auto-generated training sets with different types of masked regions. The highest scores are underlined. [%]

	$T_a^{(He,5)}$	$T_a^{(He,6)}$	$T_a^{(He,7)}$	$T_a^{(He,8)}$
Ours(T_d^{people})	<u>64.35</u>	<u>66.84</u>	<u>72.96</u>	<u>79.47</u>
Ours(T_d^{rect})	58.80	59.41	60.70	61.84
Ours($T_d^{people(ad)}$)	57.72	57.35	58.31	59.74
Ours(T_d^{all})	61.96	63.60	67.75	74.47

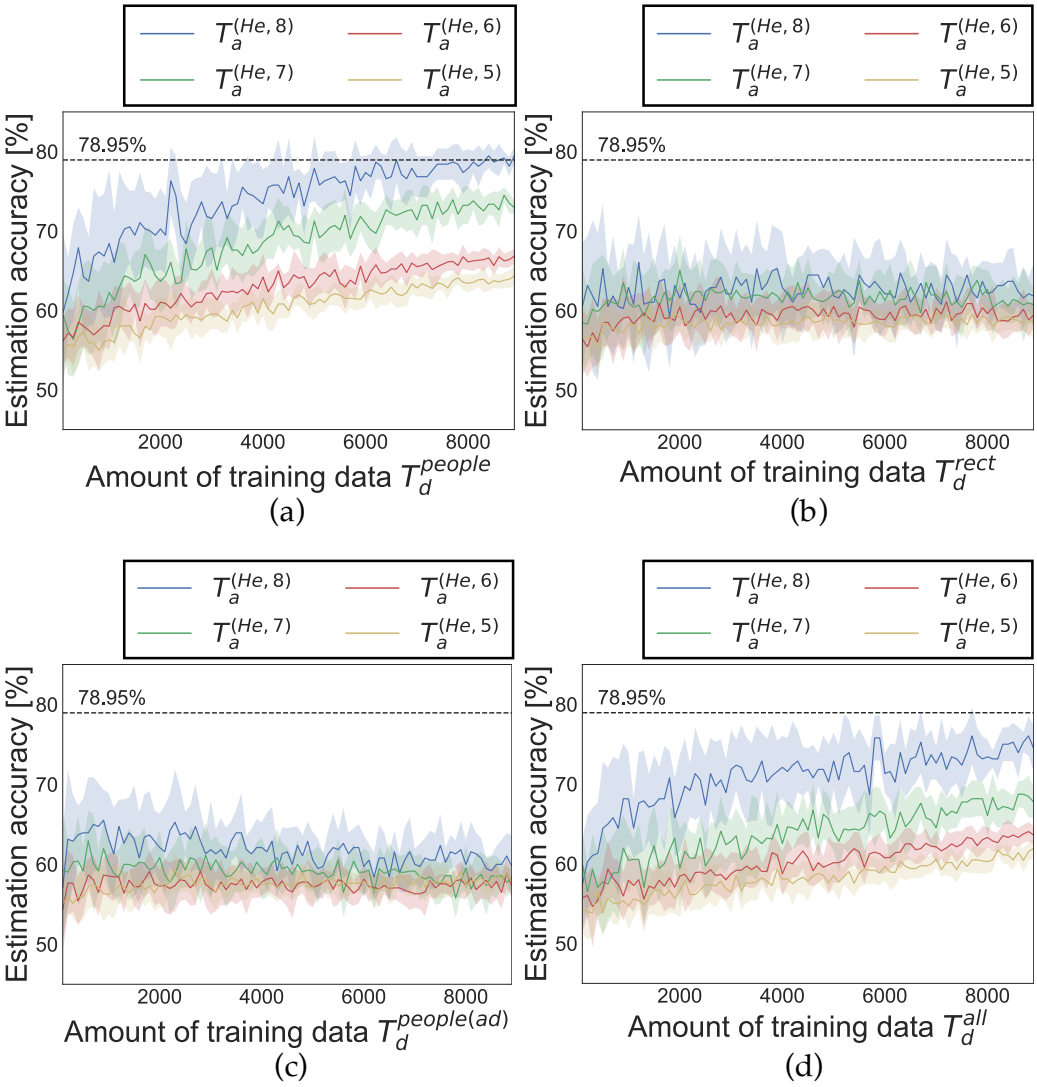


Figure 3.18: Prediction accuracy for each T_a^S (S=5,6,7,8) with (a)T_d^{people}, (b)T_d^{rect}, (c)$T_d^{people(ad)}$, and (d)T_d^{all}

accuracy is an average of 10 runs with standard deviation. As reference, black dotted lines indicate estimation accuracy with auto-generated data with manually designated masked region, i.e., T_d^{He}, which is considered to be the most effective training data proposed in 3.3.3. Table 3.8 compares estimation accuracy for each T_a^S with four types of training sets.

As shown in the graphs in Fig. 3.18 and Table 3.8, estimation accuracy with T_d^{people} was 79.47%. Although this training set requires no human labor for both masked region designation nor training data generation, the estimation accuracy was rather higher than that of T_d^{He} described in 3.5.2.1, which requires annotated masked regions. Regarding other types of masked regions, i.e., T_d^{rect}, $T_d^{people(ad)}$, and T_d^{all}, the estimation accuracies were 59.74 to 74.47, which were far below the accuracy with T_d^{people}. These results support our assumption that designation of

Figure 3.19: Failure case of our masked region generation. Since the people's region is protruded from masked region, multi levels of auto-generated data are biased towards worse quality direction.

the masked region is important to generate effective auto-generated training data.

Figure 3.17 explains possible reasons for the accuracy with T_d^{rect}, $T_d^{people(ad)}$, and T_d^{all} did not reach that of T_d^{people} with failure examples of auto-generated data. The failure case in (b) shows an example where the masked people's region is adjacent to another people's region, and the failure case shown in (c) shows an example where the object's region to be inpainted is revealed from the mask. In either case, since the masked regions are not appropriate for inpainting, auto- generated data results with degradation level $L = 0$ have lower quality, even though ideally they should not include any degradation. Therefore, we consider that the effectiveness of training data might be reduced because the quality of each level of simulated data was biased towards the worse quality direction and it was difficult to get definitive preference relationships between them.

In addition, although auto-generated training set T_d^{people} with proposed masked region showed higher accuracy than T_d^{He}, which requires annotated masked regions, we consider that there is still room for improvement. Our masked region generation strategy avoids adjacency with other objects in images and protruding object regions. However, we found that it was not always satisfied. Figure 3.19 shows examples of failure cases of our mask generation. The masked people's region is protruded from the masked region and degraded inpainted images are biased towards worse quality direction. This is the current limitation of our proposed method. We are planning to optimize the masked region towards effective inpainting as a subject for future work.

3.6 Conclusion

This chapter describes a learning-based ranking framework for image inpainting. Unlike existing learning-based IQA methods, our method trains without using subjectively annotated data by using auto-generated data; we used simulated "failed" inpainted images by focusing on inpainting algorithms. In addition, we also proposed an automatic masked region generation method for auto-generated

data, with the aim of generating completely effortless training data. Preference order estimation experiment results suggest the method's efficacy and several investigations suggest the validity of using auto-generated data instead of subjectively annotated data.

In future work we will optimize the balance between the amount of auto-generated data and their quality for our proposed system and optimize masked region towards more effective masked region generation. Applying neural network (NN)-based rank learning is also for our future work. Since our proposed data-generation method can increase the amount of training data, we argue that the NN-based ranking algorithm has a high affinity with our method. Also, we believe that the idea of generating training data by daringly generating failed images can be widely applied to other tasks requiring subjective evaluations such as image colorization [52] or image transfer [53]. Investigating the efficacy for these other tasks is also a subject for our future work.

Chapter 4

Mask Optimization for Image Inpainting

In this chapter, I and contributors introduce a novel approach of image inpainting that optimize the shape of masked region given by users. In image inpainting, users draw a mask to specify the region. However, it is widely known that users typically need to adjust the mask region by trial and error until they obtain a desired naturally inpainting result, because inpainting quality is significantly affected by even a slight change in the mask. This manual masking takes a great deal of users' working time and requires considerable input. Thus, to reduce such human labor, we propose the method for masked region optimization so that good inpainting results can be automatically obtained. To this end, our approach estimates "naturalness of inpainting" for all super pixels in inpainted image and reforms an original mask on the super-pixel-basis so that the naturalness of the inpainting result is improved. The efficacy of this approach does not depend on inpainting algorithms, thus it can be applied for every inpainting method as a plug-in. To demonstrate the effectiveness of our approach, we test our algorithm with varied images and show it outperforms existing inpainting methods without masked region reformation.

4.1 Introduction

Many effective inpainting algorithms have been researched [6, 8, 9, 10, 11], and recently researchers have tried to apply convolutional neural networks (CNNs) to this domain, and in doing so showed great improvement could be obtained in the inpainting performance [54, 55, 56, 13, 57]. However, all the inpainting algorithms reported so far share the same limitation.

They assume that a user manually draws a mask to specify an unwanted region as an offline process; they mask either by drawing the boundary [58, 59, 60] or marking few points on the target region [61, 62], or marking few scrambles on the region [63, 64]. The inpainting quality thus significantly depends on the masked region.

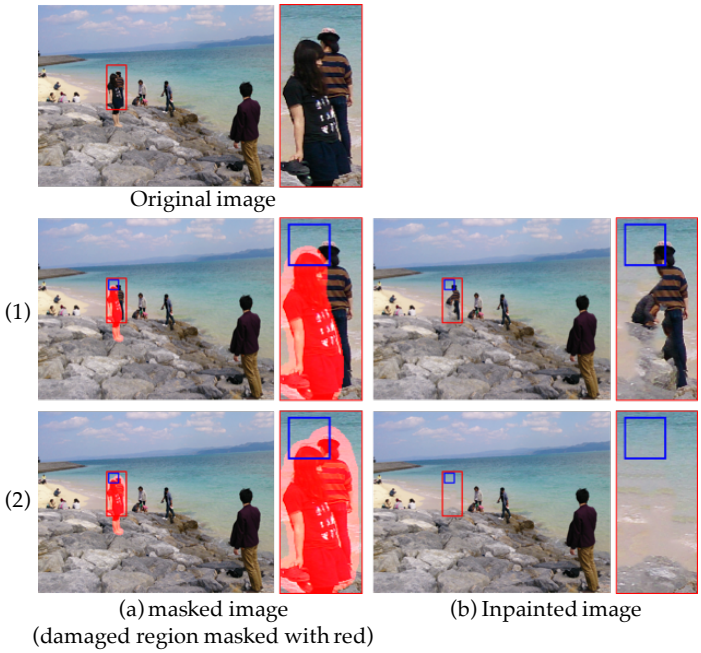


Figure 4.1: An example that shows how masked region reformation affect total inpainting quality. Although inpainted result is unnatural with original masked region (1), reformed masked region (2) achieves better inpainted result.

To explain this bottleneck and achieve better inpainting, we go back to the workflow of inpainting with the basic patch-based approach. It consists of two parts: (1) users mask unwanted regions in the input image and (2) users execute the inpainting process for the masked image. Existing inpainting methods focus only on the latter part.

To review how the masked region affects inpainting quality, we consider an inpainting task to remove the image of a woman standing in front of the image of another woman with two different masked regions as shown in Figure 4.1. Hereinafter, we focus on the patch-based approach, while the discussion here does not lose the generality. In Figure 4.1, both (1-a) and (1-b) and (2-a) and (2-b) represent a masked region and its inpainted results. Interestingly, although it is typically expected that better inpainting results should be obtained for (1-a) because the masked region is smaller, they are actually less natural.

The reason differences due to the masked region occur is shown with by blue patches in the Figure 4.1. The patch-based approach divides the original image into small patches for replacing a patch containing masked region with a similar patch only containing source region. Note that the source region consists of pixels that do not belong to the masked region. The similarity is computed merely by using the source region in the patch. In the image shown in (1-a) there are no similar patches to the blue patch since the latter includes a unique texture, i.e., that of a part of the neighboring person's body.

Patch retrieval failure deteriorates inpainting quality as shown in Figure 4.1(1-b). In contrast, with the blue patch in (2-a), which includes only sea textures outside of the mask textures, many similar patches exist. Thus, the inpainting quality of (2-b) becomes better though the source region is smaller. As shown in this example, the masked region should be optimized to achieve perceptually natural inpainting results. Although one might think the best mask region could be acquired by perfectly segmenting the object region, a desired inpainting result is not always generated in the manner discussed above.

This chapter proposes a masked region optimization method. The method reforms an initial masked region drawn by a user towards perceptually natural inpainted results. In particular, the method iteratively localizes unnaturally inpainted regions and reforms the masked region so that the localized regions do not form the contour of masked region. There are two technical issues to achieve this: (1) localizing unnaturally inpainted region, and (2) reforming the masked region.

For the former issue, we consider applying an image quality assessment (IQA) technique for image inpainting [3]. The original method applies a learning-to-rank approach to judge which inpainting result is more natural given two inpainted results. However, it does not localize unnatural regions (in this thesis we define "unnatural" as "unnatural in human-perception occurred by inpainting failure") in all images nor indicate where reformation should occur in the masked region. Thus it cannot be directly applied to our method. For the latter issue, we should consider an efficient strategy for reforming masked regions regarding computational cost. Since unnatural region localization should be performed for all masked regions, a huge number of iterations (masked region reformation, inpainting, and assessment) would be required if we reformed the masked region on a per-pixel basis.

Therefore, to address both the former and latter issues, we applied the super pixel concept [65]. This is an entity that groups similar pixels given a specified criteria. It can be used as computation units of localized unnaturalness and reformation to reduce the computational cost. In the work we report in this chapter, we extended IQA methods for inpainting so that they locally assess the naturalness of super pixels rather than entire images. Then we dilated or eroded masked regions so that the super pixels with unnaturalness do not form the contour of masked region.

Contributions: The main contribution of this chapter is proposing masked region optimization, a new solution to improve image inpainting. We also propose a super-pixel-wise unnatural region localization and masked region reformation algorithms to verify the validity of the main contribution.

The rest of this chapter is organized as follows. In Section **4.2** we briefly review related work. Section **4.3** describes the super-pixel-based mask optimization approach we propose. In Section **4.4**, we verify the method's efficacy with varied

experiments. Section **4.5** reviews and discusses experimental results, and in Section **4.6**, we conclude the chapter with a summary of key points and a mention of future work to be done.

4.2 Related work

This section first introduces existing work for image inpainting in subsection 4.2.1. Then, we review existing methods to solve the two remaining issues we described in the previous section. Subsection 4.2.2 shows IQA methods for naturalness estimation, and then subsection 4.2.3 introduces super pixels as the reformation unit.

4.2.1 Image inpainting methods

Many effective approaches for image inpainting have been proposed [66]. Exemplar based inpainting methods that fill unwanted regions by using other regions in images or databases are acknowledged as a promising approach. Criminisi et al. [6] first proposed exemplar-based inpainting based on patch retrieval. This algorithm replaces a target patch with patches similar to it from source regions in images. However, this method fails if there are no appropriate patches in the source region. To overcome the limitation, approaches that increase the patch availability have been proposed. There are mainly two approaches: transforming patches or retrieving patches with relaxed constraints. Patch transforming approaches use patches unsuitable for filling holes in their original condition by transforming the patch geometry. Darabi et al. [9] introduce scaling and rotation of patches while Huang et al. [11] allow projective transformation. As an approach with relaxed constraints, it was found that retrieving patches in different feature space makes restoration more effective than in original spaces such as motion field [67] and lower dimensional space [18].

CNN-based inpainting methods have also been proposed [54, 55, 56]. Yang et al. [57] extended CNN-based inpainting to larger masked regions. They proposed a context encoder to learn features by inpainting based on GAN. Iizuka et al. [13] proposed locally and globally consistent inpainting based on GAN. To train the network, they use global and local context discriminators to distinguish real images from completed ones. Although many effective methods have been proposed as described above, no inpainting algorithms have shown to be successful if the masked region is not appropriate for the inpainting task. In addition, even subtle changes in masked regions generate huge differences in inpainted results as we showed in Figure 4.1. In our work, we overcome this bottleneck by optimizing masked regions.

(a) Original image (b) Original image with masked region (c) Inpainted image 1 (d) Inpainted image 2

Figure 4.2: An example that shows the difficulty in evaluating inpainted images objectively. (a) and (b) are the original image and the masked region. (c) and (d) are inpainted images for (b) with different algorithms. Although both results are different from the original, they are perceptually natural.

4.2.2 IQA methods for inpainted images

Assessing naturalness of inpainted images is acknowledged as a task that can only be done by subjective judgment. One primary reason is explained by using Figure 4.2. In the figure, (a) and (b) respectively show an original image and one with a masked region, while (c) and (d) are inpainted images for (b) with different algorithms. Although both of these results are different from the original image, these results are perceptually natural. In the inpainting context, these results are "correct."

Existing IQA methods have tried to find a way to represent subjective quality of naturalness of inpainted results by means of objectively measurable indicators. Venkatesh and Cheung used observed gaze density inside and outside the masked region in inpainted images [23]. Instead of observed gaze, many IQA methods use a computational visual saliency map, which simulates human gaze density [22, 24, 25, 26, 27]. However, actual human gazes vary by individual viewers and viewing contexts and their correspondence with saliency maps is quite limited. Some recent saliency models are robust to general image degradations such as blurring, down-sampling, or compression-noise [68, 69]. However, they are not dedicated for finding the unnaturalness in inpainted images.

Thus, to estimate the unnaturalness of inpainting, machine learning based IQA methods have been developed [26, 27, 3]. Frantc et al. [26] and Voronin et al. [27] proposed support vector regression (SVR)-based approaches. Isogawa et al. [3] achieved more accurate subjective unnaturalness estimation for inpainted images by dividing the problem into a set of pairwise preference order estimation tasks and using the learning-to-rank approach, whose concept has been widely applied (not limited to image quality) to various tasks requiring subjective judgments [28, 29, 30, 31].

The method focuses on estimating preference orders rather than absolute scores. Here, the preference orders represent which inpainted images are more preferred (i.e., natural) by human perception. Preference orders allow us to select the best one from multiple inpainting results. The important advantage of a learning-to-rank-based approach is that it can learn only on the basis of rank order. In our

work we used this learning-to-rank-based IQA method [3] as an optimization indicator.

4.2.3 Super pixel

The "super pixel" concept, originally developed by Ren et al. [65], is a perceptually meaningful entity that groups similar pixels into smaller regions. Such super pixels have many desired properties. By grouping the pixels, super pixels reduce computational complexity [70]. They also reduce processing complexity; they carry more information than pixels and thus are perceptually meaningful objects, having the scale between the pixel level and the object level [71, 72].

Currently many applications have been proposed on the basis of such properties of super pixels [73, 71, 72]. Lucchi et al. [74] use super pixels for image segmentation to reduce computational cost and enforce local consistency. Super pixels have also been effectively used with tracking tasks [75, 76]. In such cases super pixels are used as the perspective representation of mid-level features. Zhang et al. [77] also proposed a super-pixel-based graphical model for remote sensing. They introduced super pixels as new basic units in conditional random field modeling.

To achieve computational and processing efficacy, we apply super pixels as a unit for unnatural region estimation and masked region reformation. Section 4.3 explains this in more detail.

4.3 Proposed method

We propose a mask optimization method for image inpainting. The key idea is that a masked region is reformed so that the regions that are unnaturally inpainted do not form the contours of the masked region. The proposed method consists of the following four steps (see Figure 4.3) and these procedure are repeated until Step 3 does not detect any unnatural regions.

Step1 Obtaining the original image and the current masked image (manually designated or updated after Step 4)

Step2 Inpainting the current masked image

Step3 Localizing the unnaturally inpainted regions

Step4 Reforming the mask depending on localized unnatural regions

So far, our algorithm only supports reformation of one direction, i.e., dilation or erosion. Neither larger nor smaller masked regions beyond those that are necessary decrease inpainting quality. Larger masked regions may overlap neighbor objects and reduce source regions used for filling holes. Smaller masked regions may reveal target objects that are desired to be removed.

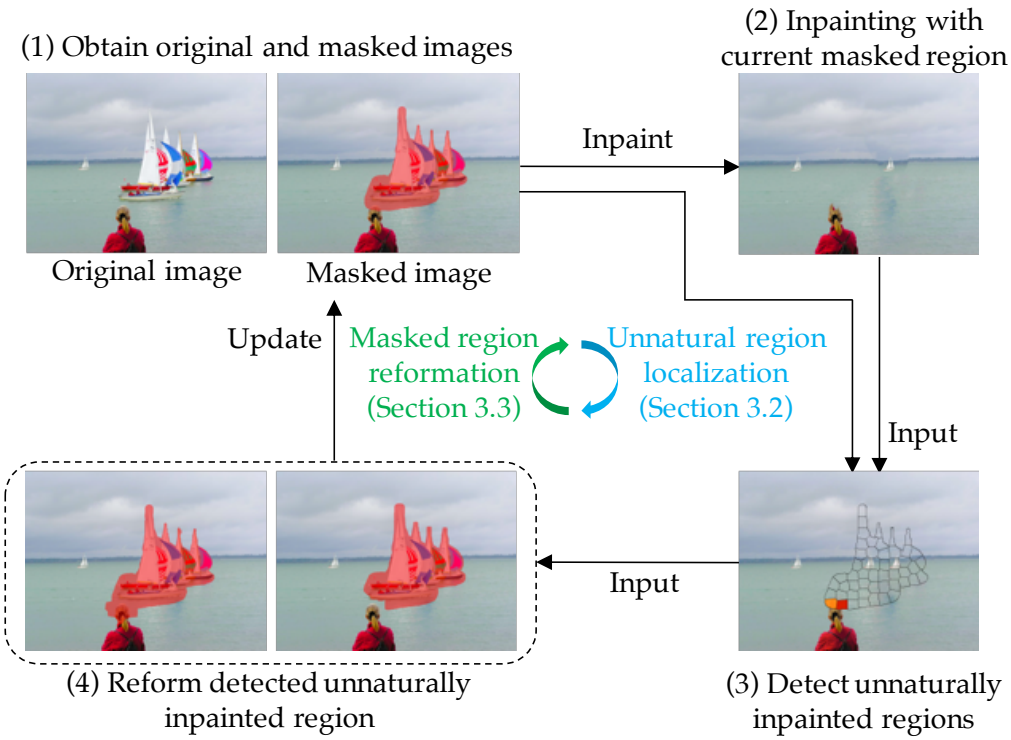


Figure 4.3: Proposed mask optimization framework overview. The method consists of the four steps; (1) obtain an original and a masked images, (2) inpaint with current mask, (3) localize unnaturally inpainted region, and (4) reform current masked region. These steps are repeated until Step 3 does not detect any unnaturally inpainted regions.

For proposed masked region optimization, we need to solve two issues. One is a way to localize unnaturally inpainted regions in Step 3. The other is a way to reform the masked region in Step 4. The following subsections first introduce the previous IQA methods for inpainted images [3] in 4.3.1, since it is the key method for our proposed method. Then, 4.3.2 and 4.3.3 respectively describe localization of unnatural regions and masked region reformation.

4.3.1 Learning-to-rank based image quality assessment

Before we describe our proposed method's details, this subsection introduces the learning-to-rank-based quality assessment for inpainted images [3], which is used in developing our unnatural regions localization. This method premises a ranking function $f(x)$ that projects inpainted images to a one-dimensional axis in accordance with unnatural inpainting.

The overview of the method's framework is shown in Figure 4.4. As training data, paired inpainted images are obtained (see (b)) with several inpainted images I^i with varied parameters as shown in (a). Then, subjective preference orders are manually annotated (see (c)) to generate inpainted pairs with preferences (see

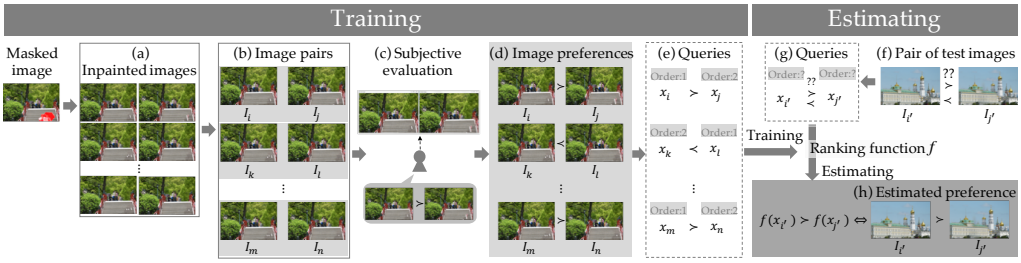


Figure 4.4: Overview of IQA method for inpainted image [3] that our unnatural region detection uses.

(d)). These samples are extracted into feature vectors x^i as shown in (e) to train ranking function $f(x)$. Given two inpainted images as shown in (f) with their extracted features (see (g)), preference orders for these two inpainted images are obtained as output values via $f(x)$. The training and estimation processes are explained below.

Train the ranking function. Hereafter, we use "$I^i \succ I^j$" to express that "I^i is preferred to I^j". We define the function $h(x^i, x^j)$ that denotes annotated preferences by subjects as follows:

$$h(x^i, x^j) = \begin{cases} +1 \ (I^i \succ I^j) \\ \ \ 0 \ (\text{no preferences}) \\ -1 \ (I^j \succ I^i), \end{cases} \tag{4.1}$$

$f(x)$ is trained so that the difference in outputs $f(x^i) - f(x^j)$ has the same sign as $h(x^i, x^j)$. In a word, the function $f(x)$ should satisfy the following formula with the training samples:

$$\text{sgn}(h(x^i, x^j)) = \text{sgn}(f(x^i) - f(x^j)). \tag{4.2}$$

The method models $f(x)$ with the linear function $f(x) = \omega^\top x$. Accordingly, Eq. 4.2 can be rewritten as

$$\text{sgn}(h(x^i, x^j)) = \text{sgn}(\omega^\top (x^i - x^j)). \tag{4.3}$$

Then, the weight vector ω satisfying Eq. 4.3 for most training data pairs is found. This is the same problem as that of binary classification. The method uses a pairwise learning-to-rank algorithm called RankingSVM [43] to solve it.

Estimate preference orders. Given pair-wise inpainted images I^a and I^b with their image feature vectors x^a and x^b, output of ranking functions $f(x^a)$ and $f(x^b)$ are calculated for all images. The preference orders between I^a and I^b are obtained as $I^a \succ I^b$ when $f(x^a) > f(x^b)$, and $I^b \succ I^a$ when $f(x^b) > f(x^a)$.

Unlike this previous IQA method for a single **whole image** evaluation, we need to **localize** the unnaturally inpainted region where requires better masked region.

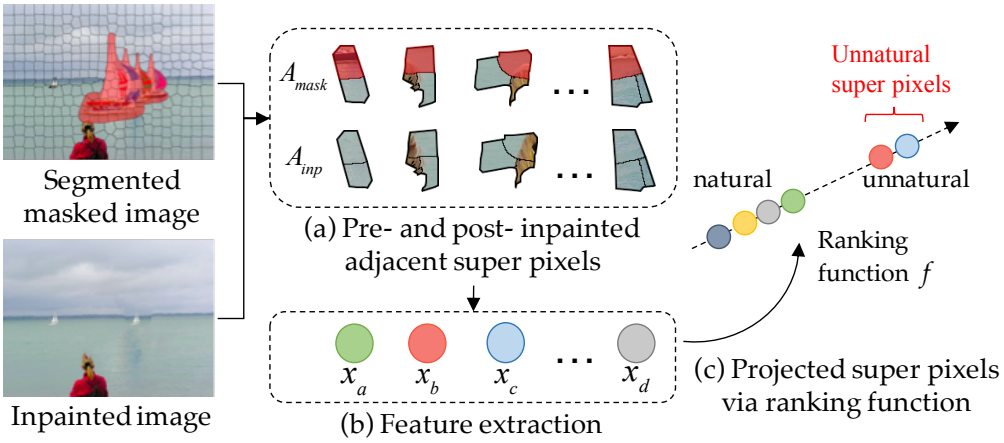


Figure 4.5: Super pixel projection to eigenspace that represents inpainting unnaturalness. With pre- and post- inpainted super pixels (a), feature vector of these super pixels are computed as shown in (b). These feature vectors are projected into an eigenspace via ranking function (c). Outlier samples in the space are detected as unnatural super pixels.

The next subsection describes how our method does it.

4.3.2 Estimating unnatural region with learning-to-rank

Now we are ready to explain how we localize unnatural regions in inpainted images. We add two improvements to the previous IQA method. First, the proposed method evaluates the relative unnaturalness for the super pixels instead of the whole images by considering a pair of adjacent super pixels as a single image. Second, it localizes unnatural super pixels by finding outlier ones in projected eigenspace that represent inpainting unnaturalness.

Applying IQA method to adjacent super pixels. We applied the previous IQA method [3] to adjacent super pixel A. We denote this adjacent super pixel as A^{mask} for a masked image and A^{inp} for an inpainted image. A^{mask} is composed of the contours of the masked region. It consists of two super pixels, one in the masked region and the other at the outside of the masked region. We denote these super pixels in the masked region as S^{mask}. Similarly, A^{inp} consists of one inpainted super pixel and the other at the outside of the inpainted region.

By considering A^{mask} and A^{inp} as pre- and post- inpainting images as shown in Figure 4.5(a), feature vectors of inpainted adjacent super pixel A^{inp} can be extracted as shown in Figure 4.5(b). Then, these features can be projected via a ranking function into a one-dimensional axis representing inpainting unnaturalness (see Figure 4.5(c)). The feature vector calculation formula is provided in Appendix A.

Localizing unnatural regions. The ranking function behavior shows that nearby coordinates are mapped on the one-dimensional axis for similar samples. That

is, the mapping via ranking function can be used to find outliers. Under the assumption that a majority of super pixel samples are naturally inpainted, outliers can be considered as unnaturally inpainted samples (see Figure 4.5(c)). Masked super pixels related with these outlier samples are detected as unnatural super pixels to be reformed.

Algorithm 1 Unnatural super pixels localization.

Input: Adjacent super pixels A^{inp} and corresponding masked super pixels S^{mask}
Output: Unnatural super pixels S^{ref}

```
N = |A^inp|
for n = 1 to N do
    x_n ⇐ CalculateFeatureVector(A_n^inp)
    f(x_n) ⇐ OutputRankingValue(x_n)
end for
F(X) = {f(x_1), f(x_2), ..., f(x_N)}
if TH has not been calculated yet then
    TH = (min(F(X)) + max(F(X)))/2
end if
for n = 1 to N do
    if f(x_n) < TH then
        Add S_n^mask to S^ref
    end if
end for
```

Detailed algorithms are as follows (See Algorithm 1). Among all adjacent super pixels in inpainted image $A^{inp} = \{A_1^{inp}, A_2^{inp}, ..., A_N^{inp}\}$ projected into an eigenspace via the ranking function, we find outlier adjacent super pixels. N is the amount of A^{inp}. To find such outlier samples with simple way to find outliers that has lower calculation cost, a threshold value TH is calculated as below.

$$TH = \frac{min(F(X)) + max(F(X))}{2} \tag{4.4}$$

where $F(X) = \{f(x_1), f(x_2), ..., f(x_N)\}$ is an ranking value vector via ranking function f for each A^{inp}. $X = \{x_1, x_2, ..., x_N\}$ represents image features for A^{inp}. TH is calculated only with the initial masked region in the first iteration and it continues to be used in the subsequent loops. Note that the output value of the ranking function does not represent the absolute score of the inpainting quality. However, the relative relationships of the output values reflect these quality orders. Therefore, we can not set the threshold value TH beforehand, and TH should be determined with relative relationships of the samples.

With the TH, masked super pixels to be reformed $S^{ref} = \{S_1^{ref}, S_2^{ref}, ..., S_M^{ref}\}$ are obtained among S^{mask} by finding corresponding outlier adjacent super pixels with lower ranking value than TH. If S^{ref} exist, the masked region is reformed; i.e., dilated or eroded. Super pixels with $f(x) < TH$ indicate more unnaturalness because we define that positive ranking values are better as shown in Eq. 4.1.

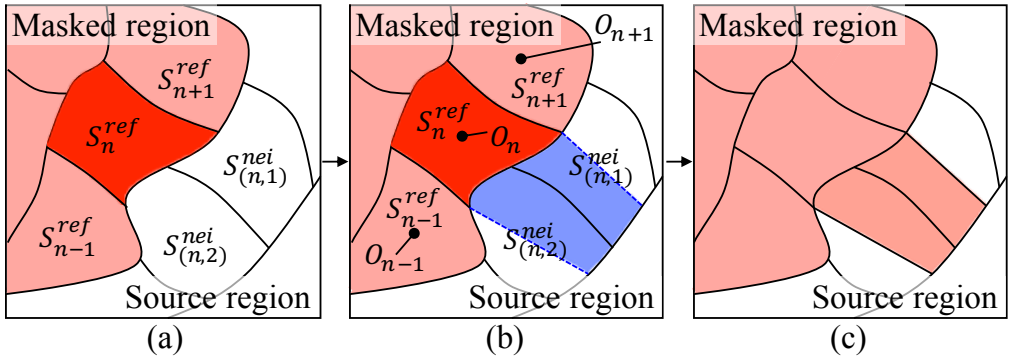


Figure 4.6: Masked region dilation processes. (a) Unnatural masked super pixel to be reformed S^{ref} and its neighbor non-masked super pixels S^{nei} are obtained. (b) To avoid over masking, not S^{nei} as a whole but only the pixels whose nearest masked super pixel is S^{ref} (shown in blue) are added to the mask. (c) Dilated masked region.

4.3.3 Masked region reformation

This subsection describes algorithms for masked region reformation. The key idea here is that masked regions are dilated or eroded so that unnatural super pixels do not form the contours of a masked region. This reforming makes it possible to avoid generating unnatural inpainting. Reforming towards dilation or erosion is decided before the optimization. In subsections 4.3.3.1 and 4.3.3.2 we respectively show masked region dilation and erosion algorithms.

4.3.3.1 Masked region dilation

The basic idea for masked region dilation is that masked regions are iteratively expanded until there are no unnatural adjacent super pixels included in A^{inp}. For expansion, a neighbor super pixel of a super pixel to be reformed S^{ref} is added into a masked region. Algorithm 2 and Figure 4.6 show the pseudo code and figures of this processing.

Let $S^{nei}_n = \{S^{nei}_{(n,1)}, S^{nei}_{(n,2)}, ..., S^{nei}_{(n,l)}\}$ be neighbor super pixels of S^{ref}_n. S^{ref}_n represents the $n-th$ super pixel to be reformed (See Figure 4.6(a)). Here, including all the pixels in S^{nei} in the masked region expands the region more than necessary. To avoid this over masking, the method selects pixels to be added into the masked region. This is because in general, smaller masked regions are better unless they do not generate unnatural inpainted regions. For each pixel p consists of S^{nei}, the closest masked super pixel is found from S^{mask}. Let the center of each S^{mask}_n be O_n as shown in Figure 4.6(b). For all pixels p, the index of the closest masked super pixel $S^{mask}_{n'}$, i.e., n' is calculated as below.

$$n' = \underset{n}{\text{argmin}}(\text{distance}(p, O_n)), \tag{4.5}$$

Algorithm 2 Masked region dilation.

Input: Masked image I_m, super pixels to be reformed S^{ref}
Output: Dilated masked image I_m

```
for n = 1 to |S^ref| do
    S^nei ⇐ neighbor super pixels of S_n^ref
    for l = 1 to |S^nei| do
        for Each pixels p consists of S^nei do
            Retrieve the nearest masked super pixel S_n'^mask
            if S_n'^mask == S_n^ref then
                Add p to masked region in I_m
            end if
        end for
    end for
end for
```

where the function *distance* calculates euclidian distance between two points.

If obtained super pixel $S_{n'}^{mask}$ is equal to S_n^{ref}, p is added to the masked region as shown in the blue region of Figure 4.6(b). After this processing for each S^{ref}, the updated masked region is obtained as shown in Figure 4.6(c).

4.3.3.2 Masked region erosion

The basic idea of masked region erosion is iterative removal of masked pixels, until there are no unnatural adjacent super pixels found from A^{inp}. For the removal, super pixels to be reformed S^{ref} are excluded while unwanted objects are not revealed. Algorithm 3 and Figure 4.7 show the pseudo code and figure of this processing.

Unlike the masked dilation algorithm, super pixels consisting of an original image are also considered. Let $S^{orig} = \{S_{(n,1)}^{orig}, S_{(n,2)}^{orig}, ..., S_{(n,l)}^{orig}\}$ be super pixels generated with original image I_o overlapped with S_n^{ref}. In Figure 4.7(a), S^{orig} are overlapped onto super pixels in masked images shown in blue dotted lines.

S^{orig} is used for deciding whether pixels are excluded from a masked region. Excluding S^{ref} as a whole may reveal unwanted objects behind the masked region. Since S^{orig} is considered to be suitable for objects in an original image, using S^{orig} can avoid unnecessary exclusion. Each pixel p of S^{ref} is excluded only when S^{orig} where p overlaps the outermost side face (See the region masked with blue in Figure 4.7(b)). After this erosion for each S^{ref}, the updated masked region is obtained as shown in Figure 4.7(c).

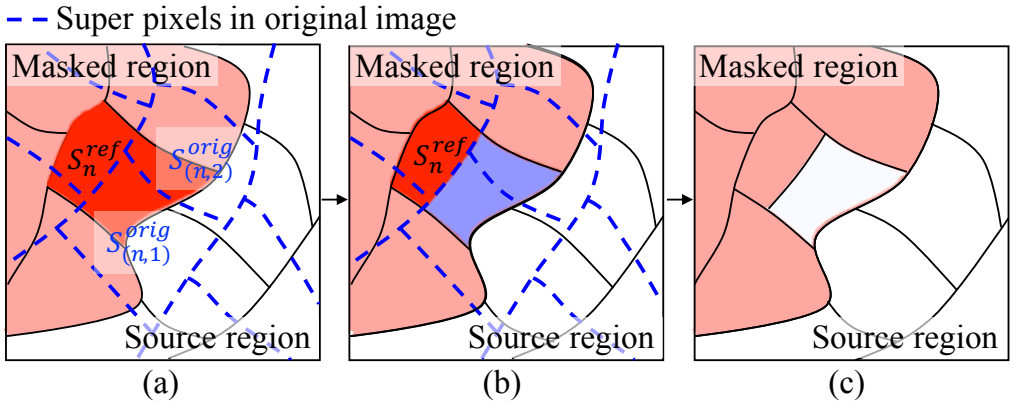


Figure 4.7: Masked region erosion processes. (a) Unnatural masked super pixel to be reformed S^{ref}, and super pixels with original image (shown with blue dotted line) that are locationally overlapped with both S^{ref} and non-masked region obtained as S^{orig}. (b) To avoid unnecessary exclusion of masks, not S^{ref} as a whole but only pixels overlapped with S^{orig} shown in blue are excluded from masks. (c) Eroded masked region.

4.4 Experiments

This section investigates the effectiveness of our proposed method. We will start with the experimental setups to obtain ranking function f in subsection 4.4.1. Then subsection 4.4.2 investigates the proposed method's effectiveness for finding unnatural inpainted images, compared with other metrics. Subsection 4.4.3 shows the effectiveness of our masked region reformation framework with various images and 4.4.4 subjectively evaluates the inpainted results.

4.4.1 Experimental setup

To generate a training set for ranking function f, 111 images with manually masked unwanted regions were prepared. The 111 images were inpainted with two existing inpainting methods [10, 11]. Six parameter sets (= 3 patch sizes $\times$ 2 levels of multi-scale parameters) were used for both methods. We randomly displayed a pair of inpainted images side-by-side as shown in Figure 4.8. Subjects were asked to choose one of three options: **r**: right image is more natural, **l**: left image is more natural, and **n**: no preference order (i.e., it is hard to decide which one is more natural). Excluding inpainted images with extremely low level of naturalness and images without consistent response of all subjects, we prepared 2,466 image pairs.

We implemented RankingSVM with SVM Rank [44] with Radial Basis Function (RBF) as the kernel function ($\gamma = 2^{-7}$), and the regularization parameter ($C = 2^{-5}$). We used a desktop PC (Intel Core i7, 3.4GHz CPU, 32GB memory) for training f. Eight subjects (four males and four females) with normal vision

Algorithm 3 Masked region erosion.

Input: Masked and original images I_m, I_o and super pixels to be reformed S^{ref}
Output: reformed masked image I_m

$S^{orig} \Leftarrow$ SuperPixelSegmentation(I_o)
for $k = 1$ to $|S^{ref}|$ **do**
  for $l = 1$ to $|S^{orig}|$ **do**
    if S_k^{ref} and S_l^{orig} are overlapped **then**
      Add S_l^{orig} to S^{orig}
    end if
  end for
  for pixels $p = (p.x, p.y)$ consists of S^{orig} **do**
    if p is included in S_k^{ref} **then**
      $I_m(p.x, p.y) \Leftarrow I_o(p.x, p.y)$
    end if
  end for
end for

Figure 4.8: Annotation interface for obtaining training data. Two different inpainted results are displayed side by side. Subjects annotate their preferences among three options: **r**: right image is better, **l**: left image is better, and **n**: no preference order.

evaluated which images were more natural. Ranking function f was trained depending on this annotation. The trained f is used for unnatural region detection in next subsection.

4.4.2 Comparison with existing metrics for estimating unnaturalness

This subsection investigates the effectiveness of our proposed unnatural region estimation for inpainted images using the trained ranking function. We compare our technique with existing IQA methods using computational saliency maps by

Figure 4.9: Stimulus images for subjective mask annotation. Top and middle rows show inpainted images. Not inpainted images are shown in the bottom row.

Hou et al. [40] and Walther et al. [42] that used by IQA methods of Voronin et al. [27] and Oncu et al. [24].

To prepare ground-truth data, we asked 12 subjects(11 males, one female) with normal vision to draw unnatural regions in 10 images consisting of seven inpainted and three not inpainted images as shown in Figure 4.9. The latter are original images to which no image processing was applied.

A drawing interface is shown in Figure 4.10. Subjects were asked to point unnatural regions out in images without any time limitation. They used a mouse as a drawing device and could change the pen size of drawing as they liked. Depending on how hard the subjects pressed the pen, the opacity (brush depth) of

Figure 4.10: User interface for pointing out unnatural region in inpainted image.

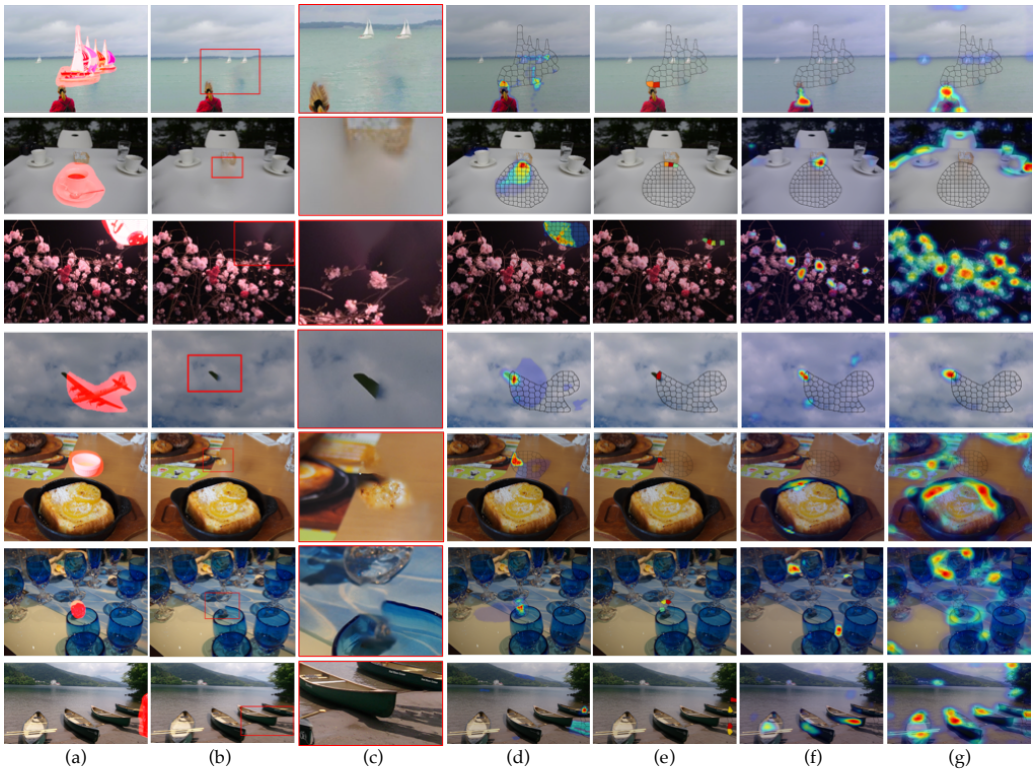


Figure 4.11: Comparison between proposed unnatural region detection and existing metrics for finding unnaturalness with subjectively annotated unnatural region as ground truth. (a) original image with damaged region masked in red, (b) and (c) inpainted image and their close-up views of unnatural regions, (d) subjectively annotated unnatural region. (e)-(g) Obtained unnatural regions with heat maps overlaid on (b) (red gathers more unnaturalness). (e) with proposed method by super pixel basis, (f) with saliency maps by Hou et al's method [40] used in Voronin et al's metric [27], (g) Walther et al.'s [42] used in Oncu et al.'s metric [24].

the line was changed. Then a heat map was generated from a drawn mask and overlaid on an image as shown in Figure 4.10. Subjects were informed that the observed images included both inpainted and unpainted images, but were not informed how many images were inpainted in order to prevent them from being affected by prior knowledge. The LCD monitor used for stimulus presentation was 21 inches (1280×1080 pixels). The distance between the monitor and the observers was 60 cm.

The inpainted image results are shown in Figure 4.11. Original images with a region masked in red are shown in (a), (b) and (c) show inpainted images and close-up views of their unnatural regions, (d) shows subjectively annotated unnatural regions as ground-truth, and (e)-(g) show computed unnatural regions as a heat map overlaid on (b) (red gathers more unnaturalness). (e) shows results obtained with the proposed method on a super pixel basis and (f) shows those obtained with the saliency map described by Hou et al. [40] and used in Voronin

et al.'s IQA metric [27]. (g) shows those obtained with the map described by Walther et al. [42] and used in Oncu et al.'s IQA metric [24].

As shown in the fourth row of Figure 4.11, for a region that is obviously unnatural such as a part of the wing of the airplane left in the uniform sky texture, all methods correctly simulate human attention. However, as shown in (f) and (g), existing saliency maps failed to simulate human attention in other rows. One of the reasons for this is that there are gaps between human gaze patterns and computational saliency maps as Isogawa et al. [3] revealed. For example, in the first and third rows, the red cloth worn by the woman or the red flowers gather more attention with existing metrics as shown in (f) and (g) because saliency maps are typically designed by assuming that warmer colors gather more gazes. In addition, in the second, fifth, and sixth rows, regions with more edges gather more attention, unlike subjectively annotated attention. The reason for this also comes from typical saliency map designs, which estimate more gazes on stronger edges. At the bottom row, existing metrics could not find unnatural inpainting because of the subtle changes in texture or color, while our metric could do so as shown in (f) and (g). As shown in (e), our method successfully estimates subjects' attention for all image stimuli, indicating that the proposed unnatural region estimation method works effectively.

4.4.3 Masked region reformation experiments

This subsection investigates the efficacy of the proposed masked region optimization method. Figures 4.12 and 4.13 respectively show masked region dilation or erosion obtained with the proposed method. In both figures, (a) shows an original image, (b) shows an initial masked region (shown in red) for the original image, (c) shows the final masked image obtained for the reformed masked region with the proposed method, and (d) shows the final inpainted images obtained with (c). Rows (e) and (f) show the output results obtained in each iterative step, and (e) shows the masked image and its close-up view with colored super pixels and their ranking value plots. Colored super pixels excluding red ones are super pixels on the contours of the masked region, which are candidates for dilation or erosion. Ranking values via ranking function f for all colored super pixels are also plotted at the bottom right of (e) in each step. All plotted samples and super pixels correspond to each other in color. Samples plotted lower than TH are outliers that indicate super pixels to be reformed. The iterations of unnatural region detection and masked region reformation empirically converges in about three to five times.

In Figure 4.12, the initial masked region shown in (b) hides an unwanted lantern, but is also overlapped with the flowers and branches of a cherry blossom tree. This provides a failed inpainted result that has discontinuities in both color and structure around the flowers or branches (See Step 1 in (f)). To avoid such failures, masked region dilation is performed. All colored super pixels other than those in red in (c) are reformation candidates. In the first iteration step, the threshold to find outlier super pixels was obtained as $TH = -0.37$. In the close-up view

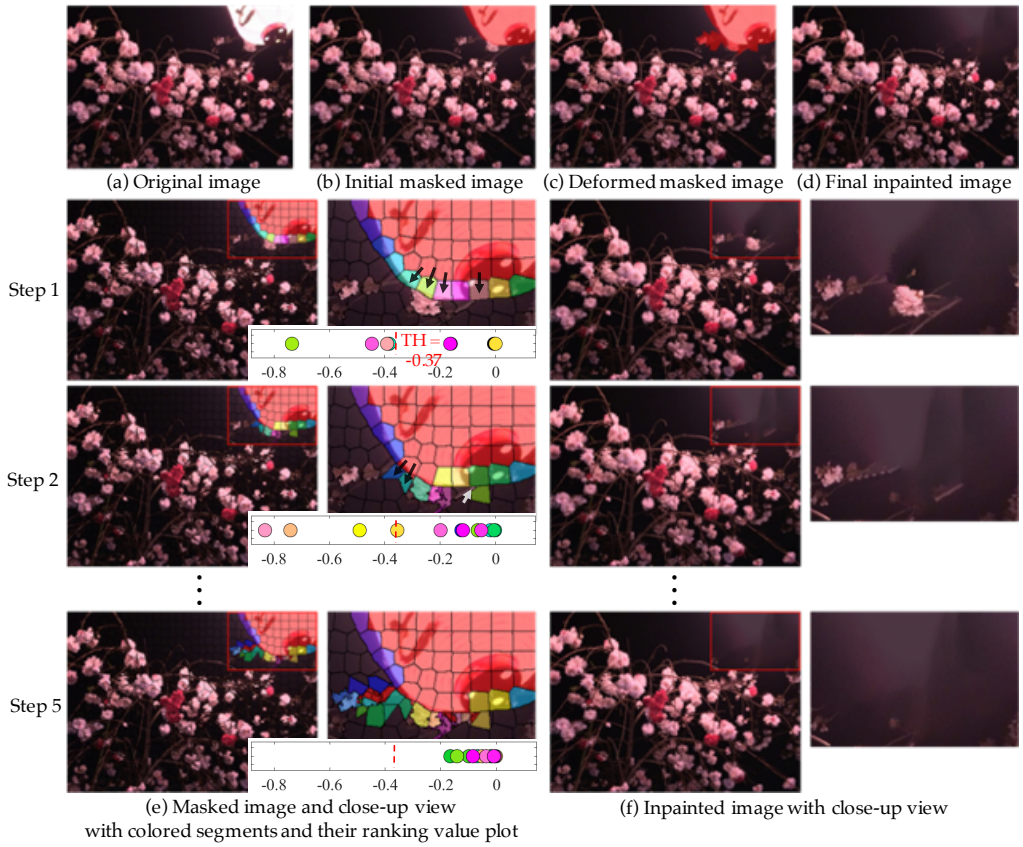


Figure 4.12: Masked region dilation result. (a) original image, (b) initial masked region (shown in red) for (a), (c) final masked image obtained for the reformed masked region, and (d) final inpainted images with (c). Rows (e) and (f) show the output results in each iterative step, and (e) shows the masked image and its close-up view with colored super pixels and their ranking value plots. Colored super pixels excluding red ones are super pixels on the contours of the masked region, which are candidates for dilation. Ranking values for all colored super pixels are also plotted at the bottom right of (e). All plotted samples and super pixels correspond to each other in color. Samples plotted lower than TH are outliers that indicate super pixels to be reformed.

in (e), outlier super pixels are annotated with the arrows and masked regions are dilated depending on such super pixels as shown in (e) in the next step. Finally, in step 5, there are no outlier super pixels and good inpainted results are obtained (See step 5 in (f)).

On the other hand, in Figure 4.13, the initial masked region in (b) masks the yachts seen above the woman. However, the region also masks the woman's head. This provides undesired inpainted results as shown in (f) at step 1, where the woman's head becomes unnaturally larger. In this case, masked region erosion works effectively. In the first iteration step, the threshold to find outlier super pixels was obtained as $TH = -0.28$. Also, in the close-up view in (e), outlier super pixels are annotated with the arrows and masked regions are eroded depending on the super pixels shown in (e) in the next step. In the final step, the

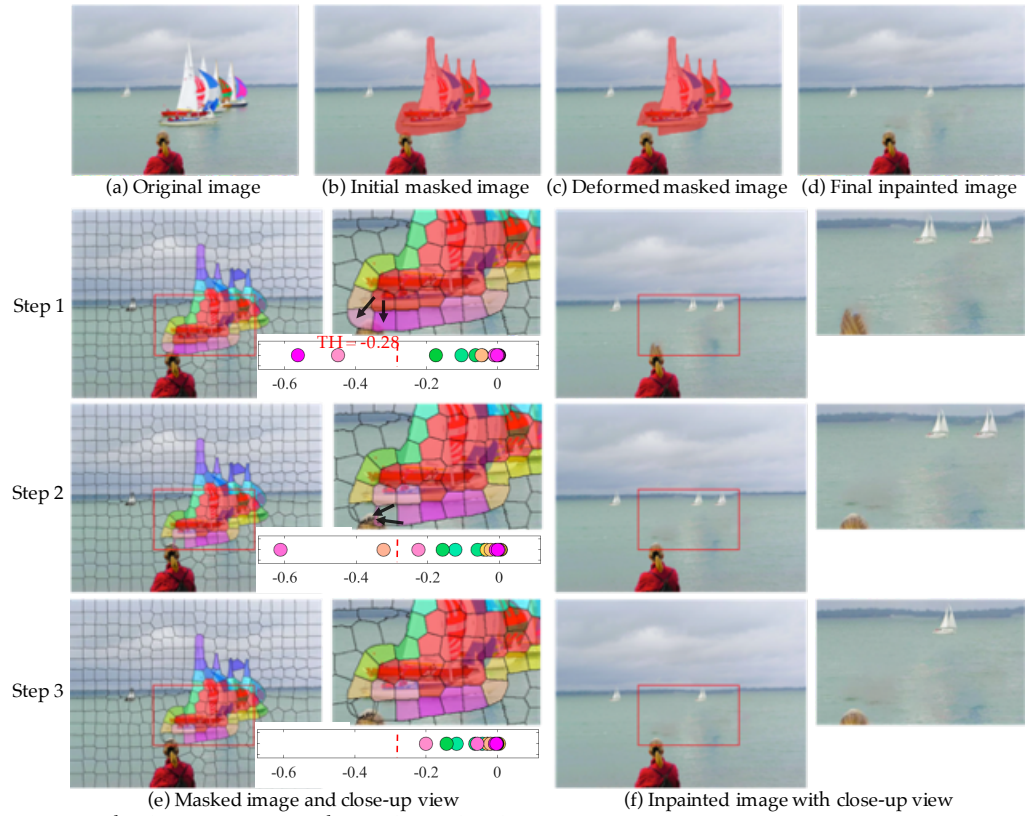


Figure 4.13: Masked region erosion result. (a) original image, (b) initial masked region (shown in red) for (a), (c) final masked image obtained for the reformed masked region, and (d) final inpainted images with (c). Rows (e) and (f) show the output results in each iterative step, and (e) shows the masked image and its close-up view with colored super pixels and their ranking value plots. Colored super pixels excluding red ones are super pixels on the contours of the masked region, which are candidates for erosion. Ranking values for all colored super pixels are also plotted at the bottom right of (e). All plotted samples and super pixels correspond to each other in color. Samples plotted lower than TH are outliers that indicate super pixels to be reformed.

masked region excludes the head region of the woman. With this masked region, an inpainted image without any unnatural super pixels is provided.

4.4.4 Subjective evaluation for inpainted image quality

This subsection subjectively compares the naturalness of resultant inpainted images depending on initial masked regions and those optimized with the proposed method. Figure 4.14 shows the test procedure, in which subjects repeated four tasks: (1) observe the first image for 10 seconds, (2) observe the second image for 10 seconds, (3) provide a score for the first image, and (4) provide a score for the second image. We asked 6 subjects (5 males and 1 females) with normal vision

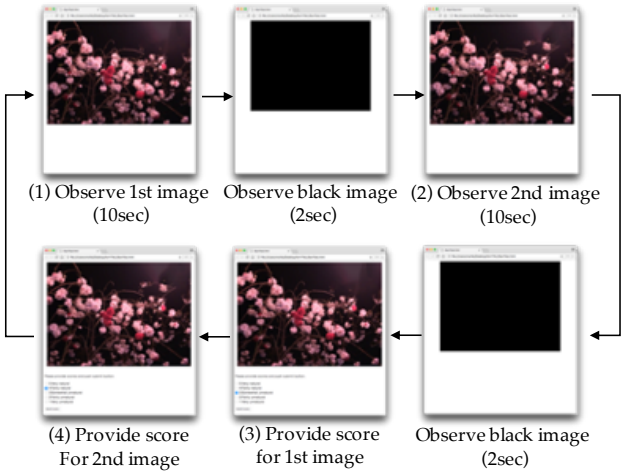


Figure 4.14: Test procedure for providing 5-point scores.

to report 5 point scores for each image (1: very unnatural, 2: fairly unnatural, 3: somewhat unnatural, 4: fairly natural, and 5: very natural). Subjects in this experiment did not duplicate those mentioned in 4.4.2. For the first and second images, inpainted images with initial and optimized masked regions are randomly shown. In order to avoid differences between two images being noticed, uniform black images are shown between tasks (1) and (2).

The image stimuli are shown in Figure 4.15. In the figure, regions masked in red and its inpainted images with close-up views for initial masked regions, while (b-1) and (b-2) show those for masked regions optimized by the proposed method. The top three rows are for masked region dilation while the bottom three rows are for erosion.

The averaged scores are shown in Figure 4.16. As shown in the figure, subjective scores for inpainted images with optimized masked regions are improved for all samples except for "snow." In the first and second rows, the initial masked region overlaps the pink flower and the branch, or wing of the airplane. These overlaps generate the texture discontinuities in the inpainted region. However, because our optimized mask includes such objects, these discontinuities are removed. In the fourth, fifth, and sixth rows, the original masked region overlaps the woman's head, the elephant's body, and the sheet on the table. These masked regions cause unnatural inpainted results due to the difficulty in finding appropriate source regions to fill in the holes. However, our optimized masked regions achieve better results by excluding such object regions from masks. In "snow", the initial masked region overlaps a red leaf and the inpainted image has discontinuity around the leaf. In this case masked region dilation was converged before the mask covers the entire leaf. This is because the edge of the texture in the inpainted region is changed gradually and generates less unnaturalness. In fact, some subjects answered that the leaves in (b-2) seemed to be buried under the snow and thus it was a natural scene.

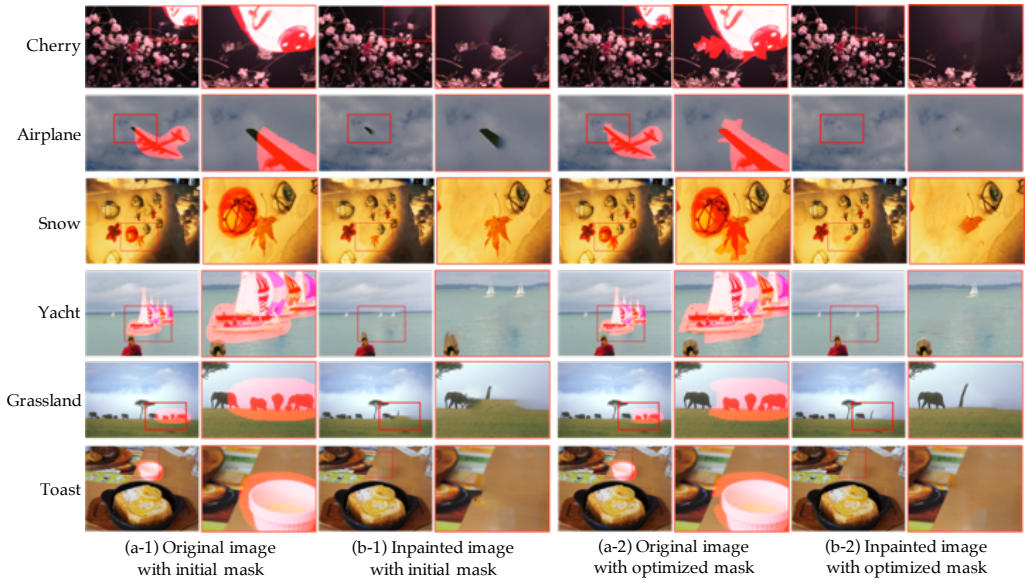


Figure 4.15: Image stimuluses for subjective evaluation.

4.5 Discussion

This section briefly reviews the experiments covered in the previous section. Unlike existing IQA methods, our method can appropriately find regions including unnatural areas in inpainted regions. This was shown through a comparison between unnatural regions that our method found and those that subjects drew. As a result of this unnatural region detection, our method effectively reformed masked regions and achieved better inpainted results. Even when inpainted results include unnatural areas, our method excludes them by dilating or eroding initial masked regions.

Here we will also mention our method's limitation. Our learning-to-rank-based unnatural super pixel detection technique depends on color and texture discontinues inside and outside damaged regions. Thus, as shown in "snow" in Figure 4.15, our natural region detection does not work well for images that are inpainted with blurred colors or textures. One possible improvement to the method is enabling it to take semantic information of unwanted objects into account.

Currently our framework outputs both dilated and eroded masked regions. This is because we cannot determine which generates better results. We believe an acceptable procedure is for users to choose one of them as a last step of the framework. However, we think the procedure in which users choose one of them as a last step of the framework is acceptable.

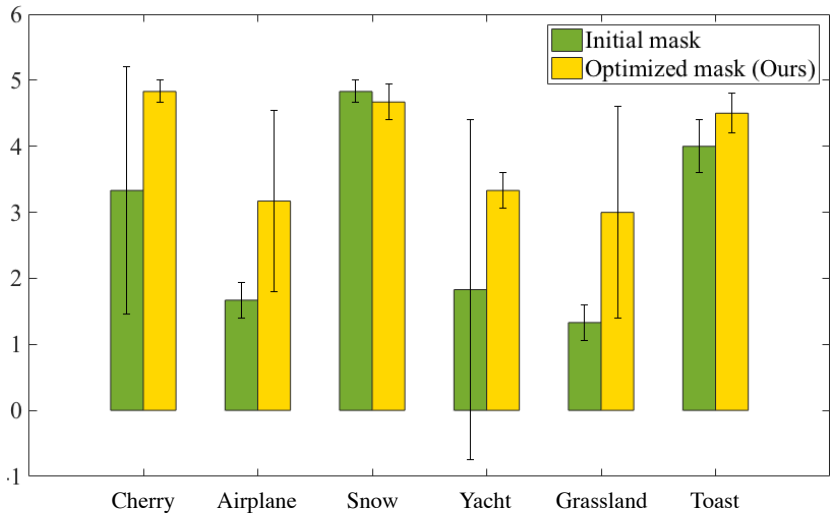


Figure 4.16: Average subjective scores for each images Figure 4.15.

4.6 Conclusion

This chapter introduced a masked region optimization framework for image inpainting. This is the first method that trials showed automatically erodes or dilates masked regions to be inpainted to achieve good inpainted results. The method also significantly reduces users' working time and the inputs they must provide because it only requires a first input of a masked region. By focusing on a learning-to-rank-based approach to estimate where unnatural inpainted results are generated in masked regions, the proposed method reforms masked regions to ease inpainting tasks. Experimental results showed that this framework effectively works.

Since our framework outputs both dilated and eroded masked regions, an interesting subject for future work will be to introduce another indicator to determine which to choose before the iterative reformation process.

Chapter 5

Inpainting via feature reduction and compensation

This chapter 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, I and contributes' 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.

5.1 Introduction and Motivation

So far, we have proposed how to tune the parameters or masked regions for inpainting. However, it is also known that even with well-tuned parameters or masked regions, inpainting often fails because of the limitation of the algorithm. With this chapter, we propose the method for improve applicability of the existing inpainting algorithms.

The most primitive solution for inpainting 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 [19]. 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 [6] was proposed, edges [78] or motions [79] 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 [15, 9, 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 [80]. 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 5.2.1.

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 $(R_1, G_1, B_1) = (200, 90, 126)$ 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. 5.1(b)): (1) converting input content to a lower dimensional feature space, (2) restoring the content in the converted lower dimensional feature space, and

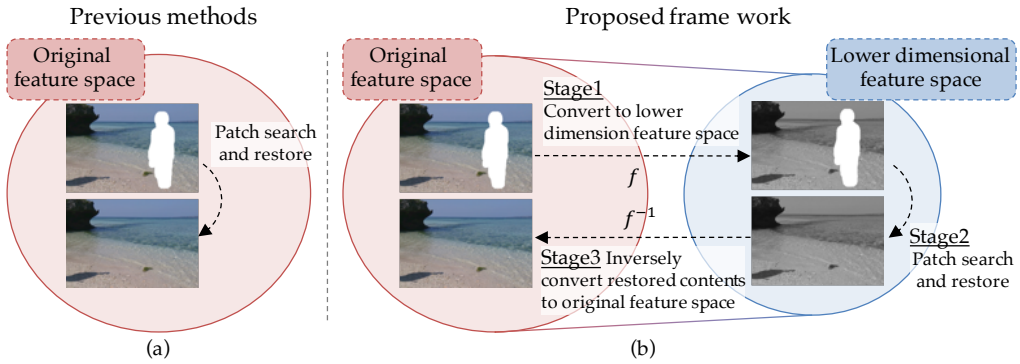


Figure 5.1: Completion process of previous methods (a) and proposed framework (b). Most previous methods complete contents in the original feature space, while the proposed framework completes them in a lower dimensional feature space. Damaged contents are converted to the lower dimensional feature space in Stage 1, restored in Stage 2, and inversely converted to the original space in Stage 3.

(3) inversely converting the restored content from the lower dimensional feature space to the original feature space.

The remainder of this chapter is organized as follows. In Section 5.2 we briefly review related work. We describe the new framework we propose in Section 5.3 and in Section 5.4 show how it works by observing image and video completion results. In Section 5.5 we show that it also works well with various feature spaces, and with some state-of-the-art completion methods. In Section 5.6, we discuss the framework's current limitations and further studies. Finally, we conclude in Section 5.7 with a brief summary.

5.2 Related work

This section reviews previous studies for content restoration and feature creation. In 5.2.1 we describe restoration methods that increase patch availability. In 5.2.2 we review methods to create and add features to the content.

5.2.1 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. [9] introduce scaling and location while Huang et al. [11] allow projective transformation; both have reported good results. Kawai et al. use patches under different illumination [15]. These methods can be implemented by allowing patch deformations or illumination changes in process step (2). However, because 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 [80]. 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.

5.2.2 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 [52] 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 [53].

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.

5.3 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 5.3.1 and then in 5.3.2 describe how to inversely convert the restored image from the lower dimensional feature space to the original feature space in Stage 3.

5.3.1 Proposed framework

Unlike existing methods, which restore damaged regions in an original feature space such as an RGB space (Fig. 5.1(a)) 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. 5.1(b). The details of each stage follow.

Stage1. 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

$$I_{in} = f(I_{in'}) \tag{5.1}$$

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 equations (5.2) and (5.3).

Note that $I_{in'_R}, I_{in'_G}$, and $I_{in'_B}$ represent each RGB channels.

$$I_{in} = f(I_{in'}) = \begin{bmatrix} 0.299 & 0.587 & 0.114 \end{bmatrix} \begin{bmatrix} I_{in'_R} \\ I_{in'_G} \\ I_{in'_B} \end{bmatrix} \tag{5.2}$$

$$I_{in} = f(I_{in'}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_{in'_R} \\ I_{in'_G} \\ I_{in'_B} \end{bmatrix} \tag{5.3}$$

Stage2. Restoration in lower dimensional space $I_{in'}$ is restored to generate I_c in the lower dimensional space.

$$I_c \longrightarrow I_{in'} \tag{5.4}$$

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.

Stage3. Inverse conversion of restored content Inverse conversion (see 5.3.2 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.

$$I_{out} = f^{-1}(I_c) \tag{5.5}$$

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 5.3.2. 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.

Fig. 5.2 explains how our framework works well. Here, Fig. 5.2(a) 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. 5.2(b). However, as shown in Fig. 5.2(c), 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. 5.2(d).

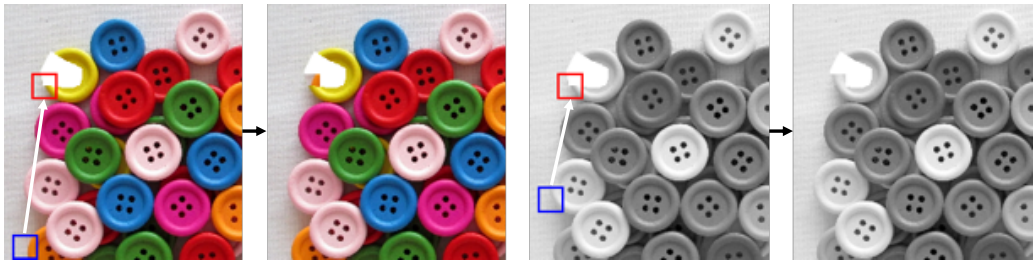


Figure 5.2: An example of how our framework contributes to a completed result, where (a) shows a damaged original image (damaged region is masked in white). In the original RGB space, an inappropriate similar patch (shown as a blue box) is retrieved for a damaged patch (shown as a red box), which results 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).

5.3.2 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 5.3.2.1 and 5.3.2.2 below.

5.3.2.1 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 [53] 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. 5.3). First, multi-scale representations of I_{in} and $I_{in'}$, before and after Stage 1, and the restored result in lower dimensional space 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}

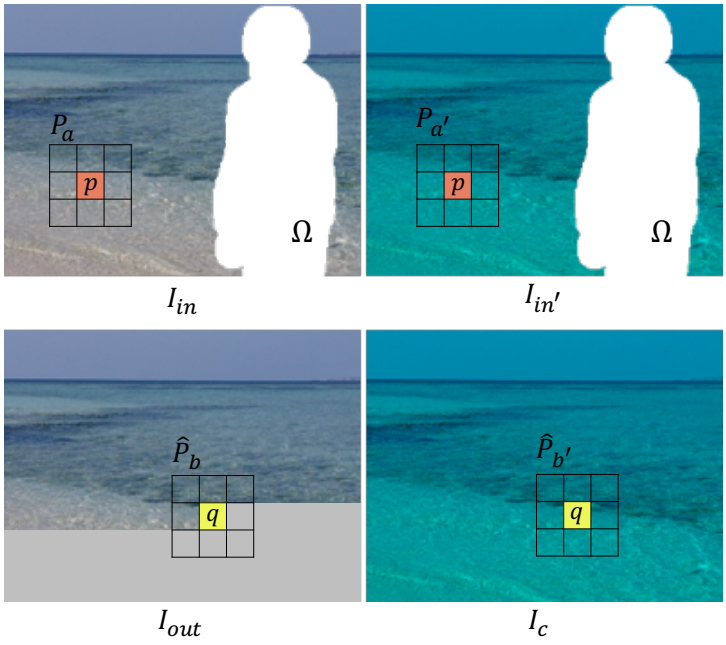


Figure 5.3: Inverse conversion from the lower dimensional feature space to the original space via versatile. Data vectors based on $P_a(p)$ and $P_{a'}(p)$, as well as patches centered at every pixel p in I_{in} and $I_{in'}$ are stocked as a database for conversion. Pixel q in restored content I_c is 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}.

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.

5.3.2.2 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 [52], 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 [52], set 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. 5.4, initial color values for colorization are set by using the color values of the original image (Fig. 5.4(a)) as follows, where

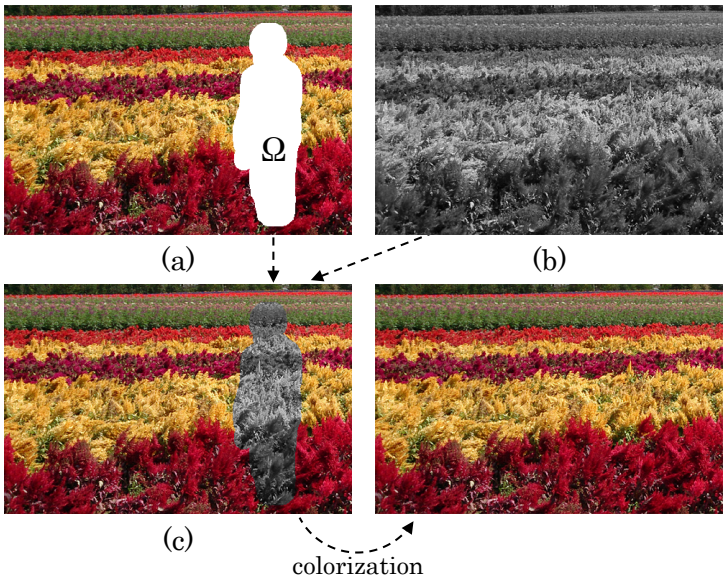


Figure 5.4: Initialization for colorization process. Initial content (c) is generated by seeding color information from non-damaged regions in I_{in} shown in (a), to restored content I_c shown in (b).

Ω represents the damaged region in the image:

$$I_c(x) = \begin{cases} I_c(x) & (x \in \Omega) \\ I_{in}(x) & (otherwise) \end{cases} \tag{5.6}$$

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.

5.4 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 5.4.1 we describe the restoration methods applied in Stage 2 and in 5.4.2 we describe in detail how our method works and show completed results with calculation times and an objective evaluation.

5.4.1 Restoration methods

For the content restoration in Stage 2 we used two methods for image [81, 6], and one method for video [82]. 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 [6] 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.

$$P(p) = C(p)D(p) \tag{5.7}$$

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

$$C(p) = \frac{\sum_{q \cap \bar{\Omega}} C(q)}{|\Phi_p|}, D(p) = \frac{|\Delta I_p^{\perp} \cdot n(p)|}{\alpha} \tag{5.8}$$

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 [81] 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.

$$\mathbf{w_{best}}(p) = argmin\, distance(P(p), \mathbf{w}(p)) \tag{5.9}$$

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 [82] 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 $\widehat{\mathbf{w}}$ 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:

$$u_p = \frac{\sum_{q \cap \bar{\Omega}} s_q u_q}{\sum_{q \cap \bar{\Omega}} s_q}, q = \{x \in \widehat{w}\} \tag{5.10}$$

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 chapter, 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)*.

5.4.2 Completion and evaluation results

In this section we demonstrate completed image results in 5.4.2.1 and video results in 5.4.2.2. In 5.4.2.3 we show objective evaluations of the methods used in the study in terms of calculation time and similarities.

5.4.2.1 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. [6] and Efros et al. [81]. Two completed target images including a complex structure and color changes were used for this comparison. The first one, shown in Fig. 5.5(I) , 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. 5.5(II) , 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. 5.5(a) 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.Crimiminisi(gray)* and *Prop.Efros(gray)* are shown in Fig. 5.5(d) and (e). Those obtained with the previous methods (Criminisi et al.'s and Efros et al.'s) are shown in Fig. 5.5(b) and (c).

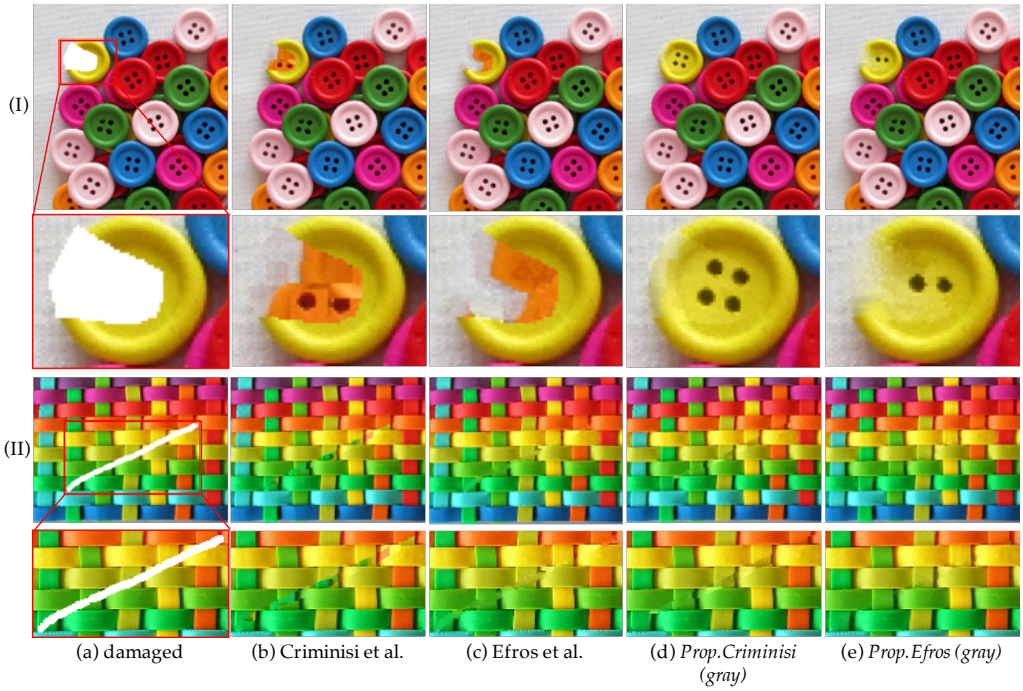


Figure 5.5: Input and result of image completion experiment; (a) Input image with damaged region masked in white. The close-up area is shown as a red frame. (b), (c) Results obtained by Criminisi et al. [6] and Efros et al. [81]. (d), (e) Results obtained with the proposed method: *Prop.Criminisi(gray)* and *Prop.Efros(gray)*.

In Fig. 5.5(a) 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. 5.5(a)) includes a rather large damaged region, it is intrinsically suitable for Criminisi et al.'s method.

From Fig. 5.5(II) 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 5.5.

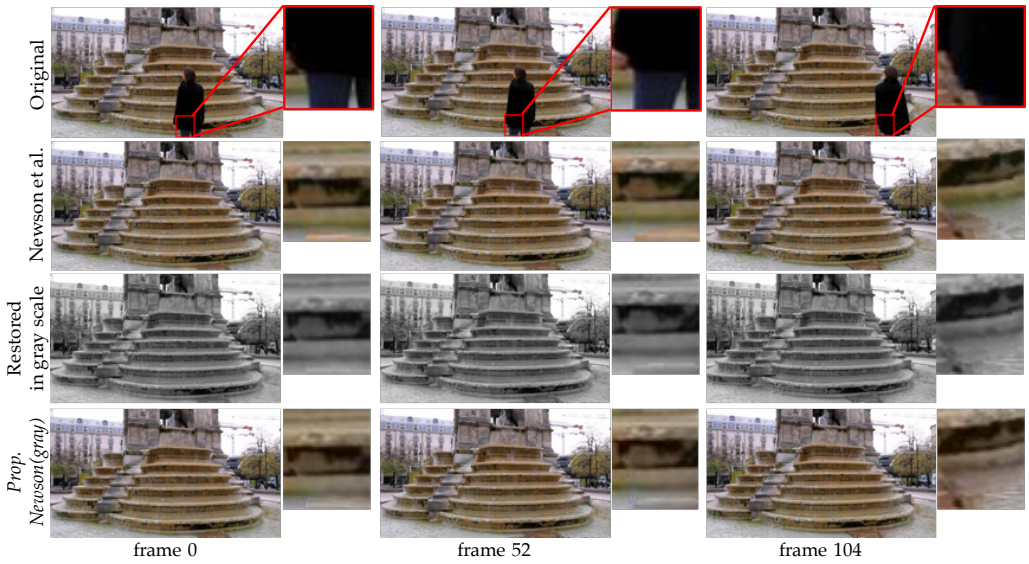


Figure 5.6: Original frames and video completion results. Original frames including an unwanted area are shown in the top row and completed results obtained with Newson et al. are shown in the second row. In our implementation, we first obtained the restored results in low-dimensional gray space (third row) and then generated the final results by colorizing them as shown in the bottom row.

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 [82] algorithm as a basis, via gray scale feature space. Fig. 5.6 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. 5.6. 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.

5.4.2.3 Objective evaluation

Objective evaluations were made among the methods in terms of similarities and computational cost. For evaluation purposes, we calculated SSIM (Structure

Table 5.1: Evaluation by SSIM with default parameters of [39]. The highest scores are underlined.

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

Table 5.2: Evaluation by PSNR[dB]. The highest scores are underlined.

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

SIMilarity) [39] and PSNR[1](Peak Signal-to-Noise Ratio) for the (I) (II) results in Fig. 5.5. 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 5.1 and 5.2. We calculated these values for original size images and also for the close-up view in Fig. 5.5. 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 5.3 shows a comparison of elapsed time between Criminisi et al.'s method and *Prop.Criminisi (gray)*, and Table 5.4 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)*

Table 5.3: Elapsed time comparison between Criminisi et al.'s method [6] and *Prop.Criminisi(gray)* [sec]. For more detail, please see 5.6.2.

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

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

Table 5.4: Elapsed time comparison between Efros et al.'s method[81] and *Prop.Efros(gray)* [sec]. For more detail, please see 5.6.2.

	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

Table 5.5: Variations of restoration methods and feature spaces. Note that all of the patterns with "*Prop.*" were obtained with our proposed methods.

restoration methods	feature spaces		
	gray scale	RG	GB
He et al [10]	*Prop.He(gray)*	*Prop.He(RG)*	*Prop.He(GB)*
Huang et al [11]	*Prop.Huang(gray)*	*Prop.Huang(RG)*	*Prop.Huang(GB)*

requires an additional process, i.e., colorization. The comparison between Efros et al's method and *Prop.Efros(gray)* (Table 5.4) 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. 5.5. To further elaborate on this point, section 5.6.2 describes how we analyzed computational cost in more detail.

5.5 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 5.5. Their details are as follows.

Restoration method. First, we introduce He et al.'s and Huang et al.'s algorithm [10, 11] for Stage2, the image restoration part.

He et al.'s method [10] 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.

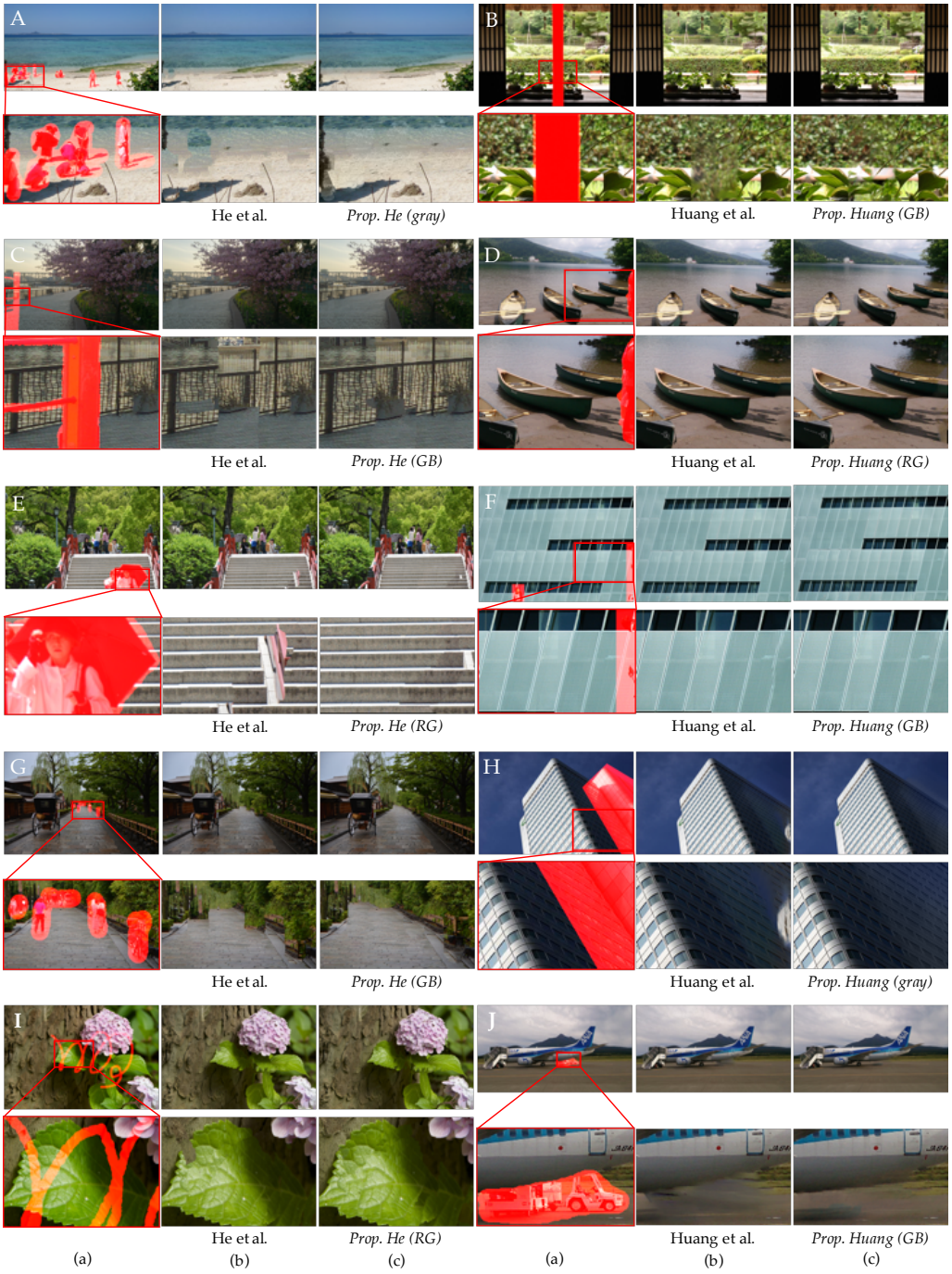


Figure 5.7: Completed results obtained with current state-of-the-art restoration methods and various feature spaces (gray, RG, and GB). (a) Target images with damaged regions (masked in red). (b) Results obtained with the original restoration method without dimension reduction. (c) Results obtained with the proposed method with dimension reduction. Note that all of the image results annotated "*Prop.*" were obtained with our proposed methods.

Huang et al.'s method [11] 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.

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 for gray scale space, we used colorization [52] in the same way as mentioned in 5.4.2. For RG space and GB space the versatile method described in 5.3.2.1 was used.

Results. Completed results for the natural scenes we used are shown in Fig. 5.7. 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.

5.6 Limitation and future work

5.6.1 Feature space selection

So far, we have showed the results obtained in using a specific lower dimensional feature space for restoration without explanation. Fig. 5.8 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.

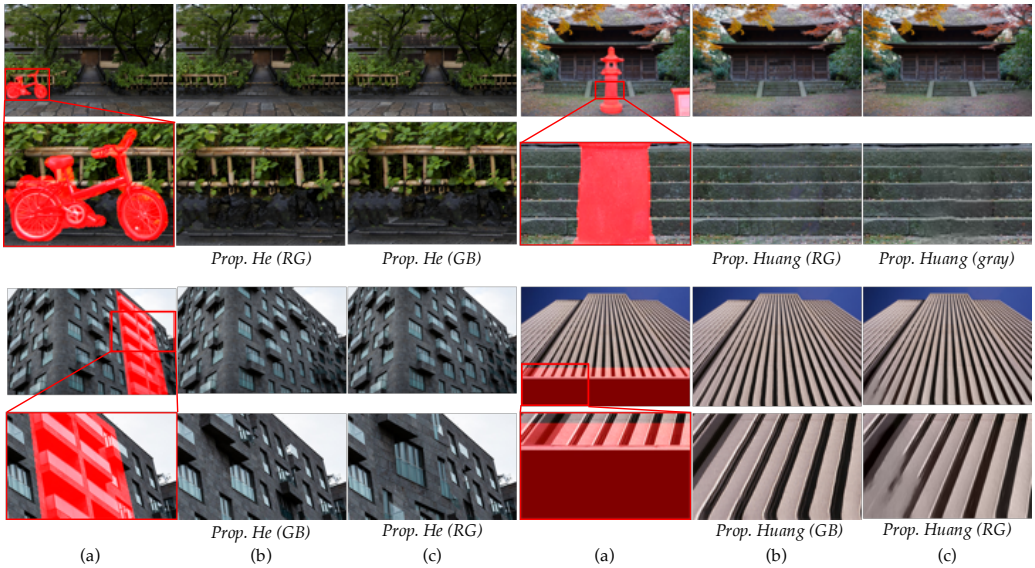


Figure 5.8: Comparison between completed results obtained using different feature spaces for restoration. Note that the same restoration method was used to obtain the two results. (a) Target images with damaged region masked in red. (b), (c) Completed results obtained with effective/ineffective feature space.

As the results show, completion performance depends on the feature space used for restoration. It was already mentioned in Section 5.4 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 [2]. 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.

5.6.2 Calculation cost

In 5.4.2.3, 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 5.3.2.2). Although we do not focus on the issue of calculation cost in this

Table 5.6: Elapsed time comparison between He et al.'s method [10] and *Prop.He(gray)* with two inverse conversion method explained in 5.3.2.1 and 5.3.2.2 [sec].

He et al.	*Prop.He(gray)*				
	Completion	inverse conversion via 5.3.2.1		inverse conversion via 5.3.2.2	
		inverse conversion	total	inverse conversion	total
19.1	14.4	1513.6	1528.0	31.6	46.0

chapter, 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 [6], 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 5.3.2. For this we introduced two approaches, a generalized approach in 5.3.2.1, and dedicated approach in 5.3.2.2. The former requires considerable computational cost. Table 5.6 shows an example of elapsed time for (b) and (c) in Fig. 5.7 A with the inverse conversion method presented in 5.3.2.1 and 5.3.2.2. 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. 5.7 A (1280×720)[pixels], the inverse conversion methods in 5.3.2.1 and 5.3.2.2 (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.

5.7 Conclusion

In this chapter, 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.

Chapter 6

Discussion and Conclusion

In this chapter, I briefly review the contributions of research work presented herein and describe future work that is beyond the scope of the contributions as presented herein. Specific discussions on details of individual approaches can be found in respective sections in previous chapters.

6.1 General Discussion

Herein, I have introduced a new approach for image inpainting in terms of "usability-awareness". Existing inpainting methods implicitly assume that the users contain considerable knowledge of image processing, which limits the range of users. However, I have considered that general users can also gain significant benefits from inpainting by reducing manual processing and improving applicability of inpainting. To this end, I have solved the problems that limit usability of inpainting (i.e., users are required to tune parameters and masked regions before inpainting). Even with well-tuned parameters and masked regions, most inpainting algorithms fail when similar regions do not exist in undamaged regions. In chapters 2-5, I have solved these important issues in the following manner:

In chapter 2, I described learning-to-rank based IQA method for image inpainting. This method solves the problem for parameter tuning by selecting the best inpainted image from several results inpainted with different parameters. I used a ranking-by-learning algorithm to estimate the order of inpainted images on the basis of subjective quality. This framework easily introduces auto-generated training data for more effective learning. In addition, image features are introduced that reflect differences around a contour of damaged regions based on the gaze measuring experiments, which indicated that a high negative correlation exists between subjective quality and gaze density around the contour. Preference order estimation experimental results suggest the method's efficacy. Especially with auto-generated training sets, and the estimation performance was about 13 percentage points higher than that of existing IQA methods.

In chapter 3, I have proposed a method for training data generation with regard to learning-to-rank-based on preference order estimation. This method covers the weaknesses of the method described in chapter 2 (i.e., the method requires huge

annotation cost for obtaining data with pair-wise preference orders). I used simulated "failed" inpainted images by focusing on inpainting algorithms. In addition, I also proposed an automatic masked region generation method with the aim of generating training data effortlessly. Preference order estimation experiment results suggest the method's effectiveness, and several investigations suggest the validity of using auto-generated data instead of subjectively annotated data.

I proposed a masked region optimization framework for image inpainting in chapter 4. This method solves the problem of users with regard to the requirement of drawing a masked region by trial and error. Since the method automatically adjusts the tuned masked region, it significantly reduces users' working time, and it only requires the first input of a masked region so the inputs must be provided. By focusing on a learning-to-rank-based approach as described in chapter 2 and 3 to estimate where unnatural inpainted results are generated in masked regions, the proposed method reforms masked regions to ease inpainting tasks. Experimental results indicated that this framework effectively improves final inpainting quality. In chapter 5, I introduced a new framework for image completion to improve the applicability of existing inpainting algorithms. Our framework involves three stages as follows: (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. The framework enhances the possibility of applying patches dissimilar to those in the original color space, and also the method makes it possible to use a variety of restoration methods and feature spaces. Experiment results have verified its effectiveness even if the images lack similar patterns to be used for restoration.

6.2 Limitation and Future Work

As described in this thesis, I explored the concept of improving image inpainting so that general users can easily deal with common issues faced as such, even if they do not have any special knowledge. While I have made significant progress toward fulfilling the objective, a considerable amount of additional research is required.

IQA with Semantics or Knowledge. Herein, I introduced IQA via learning-to-rank in chapter 2 and 3 with image feature patch-based contour consistency (*PBCC*). Although this local image feature works well, it does not consider global information. For human perception, I have hypothesized that IQA requires semantic information and previous knowledge that differs from person-to-person. Introducing such global features should be addressed in future research as a common issue for IQAs.

Education or Guide for Users toward Effective Masking. As described in chapter 4, tuned masked regions are required for successful inpainting, even though it is quite difficult for general users. To address this issue, I have proposed

masked region optimization, which is a novel approach for improving inpainting. However, I considered that other approaches also have the ability to solve this issue. For example, education of general users so that the users can learn how to mask appropriately or visual guidance of the user's masking processes are also possible approaches.

Real Time Effective Inpainting. Although many inpainting methods including our method one described in chapter 5 have been proposed, real time inpainting processing is still difficult for complex scenarios. Recently, inpainting has also been used in AR/MR applications such as Diminished Reality. Based on the results of our study, faster and more robust inpainting improves quality of such applications and expands the application of inpainting.

6.3 Conclusion

Image inpainting is a technique that automatically removes and restores unwanted regions in images. The technique has been used in various applications, and many effective algorithms have been researched so far. However, each process included in inpainting (i.e., parameter tuning, masking unwanted region, and restoration) contain difficulties for general users, which is described in chapter 1.

Herein, I have solved each problem using the following four approaches: preference order estimation via learning-to-rank (chapter 2), training data generation for it (chapter 3), masked region optimization (chapter 4), and restoration via lower dimensional feature space (chapter 5).

By approaching all the steps included in the workflow, I believe that this thesis comprehensively improves the applicability of image inpainting. In particular, I aim to enhance the ease of using the use of inpainting even for general users, who do not have specific knowledge of image processing.

Appendix A

Image features for learning to rank

In chapter 4 in this thesis, as image features x dedicated for evaluating inpainted images, we used the 10-dimensional vector $x = (X_d, X_s)$, where X_d and X_s represent unnaturalness produced by color or structural discontinuity in an image. All of X_d and X_s have 5-dimensional values. X_d and X_s are computed as below;

$$X_d = ||S(P_{in}) - S(P_{out})||_2^2 \tag{A.1}$$

$$X_s = \frac{\sum_{p \in \delta\Omega} S(P_{out}(p))}{\sum_{p \in \delta\Omega} 1} \tag{A.2}$$

where Ω and $\delta\Omega$ respectively denote a masked region and its contour. Eq. A.1 represents a squared 2-norm. $P_{in}(p)$ and $P_{out}(p)$ show masked and source regions in patch $P(p)$, which is centered at point p. In addition, $S(P_{in}(p))$ and $S(P_{out}(p))$ represent average features of $P_{in}(p)$ and $P_{out}(p)$ as shown below.

$$S(P_{in}(p)) = \frac{\sum_{q \in P(p) \cap \Omega} s(q)}{\sum_{q \in P(p) \cap \Omega} 1} \tag{A.3}$$

$$S(P_{out}(p)) = \frac{\sum_{q \in P(p) \cap \bar{\Omega}} s(q)}{\sum_{q \in P(p) \cap \bar{\Omega}} 1} \tag{A.4}$$

In the work we report in this chapter, we used $s(p) = (u(p), v(p))$, where $u(p) = (u_R(p), u_G(p), u_B(p))$ and $v(p) = v_x(p), v_y(p)$, each denoting RGB pixel values and two-dimensional edge texture features.

Bibliography

[1] Marcelo Bertalmío, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. "Image inpainting". In: *Proc. ACM SIGGRAPH*. 2000, pp. 417–424 (cit. on pp. i, 1, 3, 76, 77).

[2] Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. "Toward Enhancing Robustness of DR System: Ranking Model for Background Inpainting". In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2015, pp. 178–179 (cit. on pp. ii, 3, 93).

[3] Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. "Eye gaze analysis and learning-to-rank to obtain the most preferred result in image inpainting". In: *IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3538–3542 (cit. on pp. ii, 3, 10, 11, 29–32, 38, 41, 42, 56, 58–62, 70).

[4] Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Hideaki Kimata. "Which is the better inpainted image? Learning without subjective annotation". In: *British Machine Vision Conference (BMVC)*. 2017, 472:1–472:10 (cit. on pp. ii, 3).

[5] Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. "Content Completion in Lower Dimensional Feature Space through Feature Reduction and Compensation". In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2015, pp. 156–159 (cit. on pp. ii, 3).

[6] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. "Region filling and object removal by exemplar-based inpainting". In: *IEEE Transactions on Image Processing* 13.9 (2004), pp. 1200–1212 (cit. on pp. 1, 3, 9, 35, 54, 57, 77, 85–87, 89, 94).

[7] Alexandru Telea. "An Image Inpainting Technique Based on the Fast Marching Method." In: *Journal of Graphics Tools* 9.1 (2004), pp. 23–34 (cit. on pp. 1, 3).

[8] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing". In: *ACM Transactions on Graphics (TOG)* 28.3 (2009), 24:1–24:11 (cit. on pp. 1, 3, 9, 54).

[9] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. "Image Melding: Combining Inconsistent Images using Patch-based Synthesis". In: *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2012)* 31.4 (2012), 82:1–82:10 (cit. on pp. 1, 3, 9, 54, 57, 77, 79).

[10] Kaiming He and Jian Sun. "Image Completion Approaches Using the Statistics of Similar Patches". In: *IEEE Transactions on Pattern Analysis and*

Machine Intelligence 36.12 (2014), pp. 2423–2435 (cit. on pp. 1, 3, 9, 14, 16, 22, 29, 35, 40, 45, 47, 48, 54, 66, 90, 94).

[11] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. "Image Completion using Planar Structure Guidance". In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)* 33.4 (2014), 129:1–129:10 (cit. on pp. 1, 3, 9, 14, 16, 22, 29, 40, 45, 48, 54, 57, 66, 77, 79, 90, 92).

[12] Jan Herling and Wolfgang Broll. "High-Quality Real-Time Video Inpainting with PixMix". In: *IEEE Transactions on Visualization and Computer Graphics* 20.6 (2014), pp. 866–879 (cit. on pp. 1, 3, 40, 45).

[13] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–14 (cit. on pp. 1, 3, 54, 57).

[14] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. "Generative Image Inpainting with Contextual Attention". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on pp. 1, 3, 40, 47, 48).

[15] Norihiko Kawai, Kotaro Machikita, Tomokazu Sato, and Naokazu Yokoya. "Video Completion for Generating Omnidirectional Video without Invisible Areas". In: *Information and Media Technologies* 6.1 (2011), pp. 158–171 (cit. on pp. 3, 77, 79).

[16] M. Isogawa, D. Mikami, D. Iwai, H. Kimata, and K. Sato. "Mask Optimization for Image Inpainting". In: *IEEE Access* 6 (2018), pp. 69728–69741 (cit. on p. 3).

[17] Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. "Image and video completion via feature reduction and compensation". In: *Multimedia Tools and Applications* 76.7 (2017), pp. 9443–9462 (cit. on p. 3).

[18] Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. "Image and video completion via feature reduction and compensation". In: *Multimedia Tools and Applications* (2016), pp. 1–20 (cit. on pp. 9, 29, 57).

[19] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. "Simultaneous Structure and Texture Image Inpainting". In: *IEEE Transactions on Image Processing* 12.8 (2003), pp. 882–889 (cit. on pp. 9, 76).

[20] Zongben Xu and Jian Sun. "Image Inpainting by Patch Propagation Using Patch Sparsity". In: *IEEE Transactions on Image Processing* 19.5 (2010), pp. 1153–1165 (cit. on p. 9).

[21] Song Wang, Hong Li, Xia Zhu, and Ping Li. "An Evaluation Index Based on Parameter Weight for Image Inpainting Quality". In: *International Conference for Young Computer Scientists (ICYCS)*. 2008, pp. 786–790 (cit. on p. 10).

[22] Paul A. Ardis and Amit Singhal. "Visual salience metrics for image inpainting". In: *Proc. SPIE*. Vol. 7257. 2009, 72571W–72571W–9 (cit. on pp. 10, 25, 29, 30, 41, 58).

[23] M. Vijay Venkatesh and Sen-Ching S. Cheung. "Eye tracking based perceptual image inpainting quality analysis". In: *Proc. IEEE International*

Conference on Image Processing (ICIP). 2010, pp. 1109–1112 (cit. on pp. 10, 12, 16, 25, 29, 41, 58).

[24] AlexandraIoana Oncu, Ferdinand Deger, and JonYngve Hardeberg. "Evaluation of Digital Inpainting Quality in the Context of Artwork Restoration". In: *European Conference on Computer Vision (ECCV) Workshops and Demonstrations*. Vol. 7583. 2012, pp. 561–570 (cit. on pp. 10, 12, 14, 15, 23, 25, 26, 29, 30, 41, 42, 58, 68–70).

[25] A.DANG Thanh Trung, B.Azeddine Beghdadi, and C.Chaker Larabi. "Perceptual quality assessment for color image inpainting". In: *IEEE International Conference on Image Processing (ICIP)*. 2013, pp. 398–402 (cit. on pp. 10, 12, 14, 15, 29, 30, 58).

[26] V. A. Frantc, V. V. Voronin, V. I. Marchuk, A. I. Sherstobitov, S. Agaian, and K. Egiazarian. "Machine learning approach for objective inpainting quality assessment". In: *Proc. SPIE*. Vol. 9120. 2014, 91200S–91200S–9 (cit. on pp. 10, 12, 22, 23, 29, 30, 32, 58).

[27] V. V. Voronin, V. A. Frantc, V. I. Marchuk, A. I. Sherstobitov, and K. Egiazarian. "No-reference visual quality assessment for image inpainting". In: *Proc. SPIE*. Vol. 9399. 2015, 93990U–93990U–8 (cit. on pp. 10, 12, 14, 15, 22, 23, 29, 30, 32, 58, 68–70).

[28] Kuang-Yu Chang and Chu-Song Chen. "A Learning Framework for Age Rank Estimation Based on Face Images With Scattering Transform". In: *IEEE Transactions on Image Processing* 24.3 (2015), pp. 785–798 (cit. on pp. 10, 12, 31, 58).

[29] Jianzhou Yan, Stephen Lin, Sing Bing Kang, and Xiaoou Tang. "A Learning-to-Rank Approach for Image Color Enhancement". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 2987–2994 (cit. on pp. 10, 12, 17, 31, 58).

[30] Takashi Abe, Takayuki Okatani, and Koichiro Deguchi. "Recognizing surface qualities from natural images based on learning to rank". In: *International Conference on Pattern Recognition (ICPR)*. 2012, pp. 3712–3715 (cit. on pp. 10, 12, 17, 31, 58).

[31] Aditya Khosla, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. "Memorability of Image Regions". In: *Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 296–304 (cit. on pp. 10, 12, 17, 31, 58).

[32] Fei Gao, Dacheng Tao, Xinbo Gao, and Xuelong Li. "Learning to Rank for Blind Image Quality Assessment". In: *IEEE Transactions on Neural Networks and Learning Systems* 26.10 (2015), pp. 2275–2290 (cit. on pp. 10, 11, 13, 22, 23).

[33] Long Xu, Jia Li, Weisi Lin, Yongbing Zhang, Lin Ma, Yuming Fang, and Yihua Yan. "Multi-task Rank Learning for Image Quality Assessment". In: *IEEE Transactions on Circuits and Systems for Video Technology* PP.99 (2016), pp. 1–1 (cit. on pp. 10, 11, 13).

[34] Long Xu, Jia Li, Weisi Lin, Yun Zhang, Yongbing Zhang, and Yihua Yan. "Pairwise comparison and rank learning for image quality assessment". In: *Displays* 44 (2016), pp. 21 –26 (cit. on pp. 10, 11, 13).

[35] Lin Ma, Long Xu, Yichi Zhang, Yihua Yan, and King Ngi Ngan. "No-Reference Retargeted Image Quality Assessment Based on Pairwise Rank Learning". In: *IEEE Transactions on Multimedia* 18.11 (2016), pp. 2228–2237 (cit. on pp. 10, 11, 13).

[36] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. "Imagenet: A large-scale hierarchical image database". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255 (cit. on p. 10).

[37] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. "Articulated people detection and pose estimation: Reshaping the future". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 3178–3185 (cit. on pp. 10, 31).

[38] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3234–3243 (cit. on pp. 10, 31).

[39] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612 (cit. on pp. 11, 89).

[40] Xiaodi Hou and Liqing Zhang. "Saliency Detection: A Spectral Residual Approach". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007, pp. 1–8 (cit. on pp. 14, 15, 68, 69).

[41] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. "Frequency-tuned salient region detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 1597–1604 (cit. on pp. 14, 15).

[42] Dirk Walther and Christof Koch. "Modeling attention to salient proto-objects". In: *Neural Networks* 19.9 (2006), pp. 1395–1407 (cit. on pp. 14, 15, 23, 68–70).

[43] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. "Large Margin Rank Boundaries for Ordinal Regression". In: *Advances in Large Margin Classifiers*. MIT Press, 2000, pp. 115–132 (cit. on pp. 17, 33, 61).

[44] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. "Large Margin Methods for Structured and Interdependent Output Variables". In: *The Journal of Machine Learning Research* 6 (2005), pp. 1453–1484 (cit. on pp. 22, 39, 66).

[45] Aude Oliva and Antonio Torralba. "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope". In: *International Journal of Computer Vision* 42.3 (2001), pp. 145–175 (cit. on pp. 22, 23).

[46] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. "Interactive Digital Photomontage". In: *ACM Transactions on Graphics (TOG)* 23.3 (2004), pp. 294–302 (cit. on p. 35).

[47] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. “Mask R-CNN”. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2017, pp. 2980–2988 (cit. on p. 38).

[48] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. “RankIQA: Learning From Rankings for No-Reference Image Quality Assessment”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2017 (cit. on pp. 41, 42).

[49] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. “Places: A 10 million Image Database for Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018), pp. 1452–1464 (cit. on pp. 47, 48).

[50] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep Learning Face Attributes in the Wild”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2015 (cit. on p. 47).

[51] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. “Imagenet: A large-scale hierarchical image database”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255 (cit. on pp. 47, 49).

[52] Anat Levin, Dani Lischinski, and Yair Weiss. “Colorization Using Optimization”. In: *Proc. ACM SIGGRAPH*. 2004, pp. 689–694 (cit. on pp. 53, 79, 83, 92).

[53] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. “Image analogies”. In: *Proc. ACM SIGGRAPH*. 2001, pp. 327–340 (cit. on pp. 53, 79, 82).

[54] Junyuan Xie, Linli Xu, and Enhong Chen. “Image Denoising and Inpainting with Deep Neural Networks”. In: *Advances in Neural Information Processing Systems* (2012), pp. 1–9 (cit. on pp. 54, 57).

[55] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. “Mask-Specific Inpainting with Deep Neural Networks”. In: *Pattern Recognition*. Ed. by Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch. Cham: Springer International Publishing, 2014, pp. 523–534 (cit. on pp. 54, 57).

[56] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. “Shepard Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems* 1 (2015), pp. 901–909 (cit. on pp. 54, 57).

[57] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. “High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4076–4084 (cit. on pp. 54, 57).

[58] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “"GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts”. In: *ACM SIGGRAPH*. 2004, pp. 309–314 (cit. on p. 54).

[59] Qiong Yang, Chao Wang, Xiaoou Tang, Mo Chen, and Zhongfu Ye. “Progressive Cut: An Image Cutout Algorithm that Models User Intentions”. In: *IEEE MultiMedia* 14 (2007), pp. 56–66 (cit. on p. 54).

[60] Eric N. Mortensen and William A. Barrett. “Toboggan-based intelligent scissors with a four-parameter edge model”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. 1999, pp. 452–458 (cit. on p. 54).

[61] Muhammad Shahid Farid, Arif Mahmood, and Marco Grangetto. “Image de-fencing framework with hybrid inpainting algorithm”. In: *Signal, Image and Video Processing* 10.7 (2016), pp. 1193–1201 (cit. on p. 54).

[62] Muhammad Shahid Farid, Maurizio Lucenteforte, and Marco Grangetto. “DOST: a distributed object segmentation tool”. In: *Multimedia Tools and Applications* 77.16 (2018), pp. 20839–20862 (cit. on p. 54).

[63] Yuri Y. Boykov and Marie-Pierre Jolly. “Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images”. In: *IEEE International Conference on Computer Vision (ICCV)*. Vol. 1. 2001, pp. 105–112 (cit. on p. 54).

[64] Vladimir Vezhnevets and Vadim Konouchine. ““ GrowCut”-Interactive Multi-Label ND Image Segmentation By Cellular Automata”. In: *Proceeding of Graphicon*. 2005, pp. 150–156 (cit. on p. 54).

[65] Xiaofeng Ren and Jitendra Malik. “Learning a Classification Model for Segmentation”. In: *IEEE International Conference on Computer Vision (ICCV)*. Vol. 2. 2003, pp. 10–17 (cit. on pp. 56, 59).

[66] Christine Guillemot and Olivier Le Meur. “Image Inpainting : Overview and Recent Advances”. In: *IEEE Signal Processing Magazine* 31.1 (2014), pp. 127–144 (cit. on p. 57).

[67] Takaaki Shiratori, Yasuyuki Matsushita, Xiaoou Tang, and Sing Bing Kang. “Video Completion by Motion Field Transfer.” In: *Proc. Computer Vision and Pattern Recognition (CVPR)*. 2006, pp. 411–418 (cit. on p. 57).

[68] Olivier Le Meur. “Robustness and repeatability of saliency models subjected to visual degradations”. In: *IEEE International Conference on Image Processing (ICIP)*. 2011, pp. 3285–3288 (cit. on p. 58).

[69] Chelhwon Kim and Peyman Milanfar. “Visual saliency in noisy images”. In: *Journal of Vision* 13.4 (2013), p. 5 (cit. on p. 58).

[70] Peer Neubert and Peter Protzel. “Superpixel Benchmark and Comparison”. In: *Forum Bildverarbeitung*. 2012, pp. 1–12 (cit. on p. 59).

[71] Kevin Smith Aurelien Lucchi Pascal Fua Radhakrishna Achanta Appu Shaji and Sabine Süsstrunk. “SLIC Superpixels”. In: *EPFL Technical Report* 149300 (2010) (cit. on p. 59).

[72] Kevin Smith Aurelien Lucchi Pascal Fua Radhakrishna Achanta Appu Shaji and Sabine Süsstrunk. “SLIC Superpixels Compared to State-of-the-art Superpixel Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282 (cit. on p. 59).

[73] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision (IJCV)* 59.2 (2004), pp. 167–181 (cit. on p. 59).

[74] A. Lucchi, D. Smith, R. Achanta, V. Lepetit, and P. Fua. “A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures

in EM Images". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2010, pp. 463 –471 (cit. on p. 59).

[75] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. "Superpixel Tracking". In: *IEEE International Conference on Computer Vision (ICCV))*. 2011, pp. 1323–1330 (cit. on p. 59).

[76] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. "Robust Superpixel Tracking". In: *IEEE Transactions on Image Processing (TIP)* 23.4 (2014), pp. 1639 –1651 (cit. on p. 59).

[77] Guangyun Zhang, Xiuping Jia, and Jiankun Hu. "Superpixel-Based Graphical Model for Remote Sensing Image Mapping". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.11 (2015), pp. 5861 –5871 (cit. on p. 59).

[78] Timothy K. Shih, Nick C. Tang, and Jenq-Neng Hwang. "Exemplar-Based Video Inpainting Without Ghost Shadow Artifacts by Maintaining Temporal Continuity." In: *IEEE Transactions on Circuits and Systems for Video Technology* 19.3 (2009), pp. 347–360 (cit. on p. 77).

[79] Yonatan Wexler, Eli Shechtman, and Michal Irani. "Space-Time Completion of Video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.3 (2007), pp. 463–476 (cit. on p. 77).

[80] Takaaki Shiratori, Yasuyuki Matsushita, Xiaoou Tang, and Sing Bing Kang. "Video Completion by Motion Field Transfer." In: *Proc. Computer Vision and Pattern Recognition (CVPR)*. 2006, pp. 411–418 (cit. on pp. 77, 79).

[81] Alexei A. Efros and Thomas K. Leung. "Texture Synthesis by Non-Parametric Sampling". In: *Proc. International Conference on Computer Vision (ICCV)*. Vol. 2. 1999, pp. 1033–1038 (cit. on pp. 85–87, 90).

[82] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. "Video Inpainting of Complex Scenes". In: *SIAM Journal of Imaging Science* 7:4 (2014), pp. 1993–2019 (cit. on pp. 85, 86, 88).

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Dr. Kosuke Sato and Assoc. Prof. Dr. Daisuke Iwai, for their keen insight, valuable guidance, scholarly inputs, and consistent encouragement. They have supported me since my undergraduate days and with regard to my research as well. I would also like to thank Prof. Dr. Youji Iiguni and Prof. Dr. Takayuki Nagai for their valuable comments and suggestions with regard to this study.

I would like to thank Asst. Prof. Dr. Parinya Punpongsanon for his valuable support in the laboratory and for his friendship. I would also like to thank the entire laboratory staff, including members and secretaries, for their support and friendship.

I am immensely grateful for the support from my supervisor at NTT Media Intelligence Laboratories, Dr. Dan Mikami. He has guided and encouraged me since I joined NTT and has contributed to this study in a profound way. This study would not have been successful without his vital support.

My sincere thanks to Prof. Dr. Shinsaku Hiura, who prompted me to join the Sato Lab. Also, I would like to thank Dr. Takeshi Fujioka, who guided me to learn computer science and was a research supervisor during my high school days. Without their great influence over me, I would not be where I am today.

I would like to convey my special gratitude to all my colleagues at NTT Laboratories. I would like to thank Dr. Akisato Kimura for his research mentoring. Also, I would like to thank Dr. Kosuke Takahashi, Mr. Akira Kojima, Dr. Hideaki Kimata, Dr. Yoshinori Kusachi, who extended their support in important ways. I gained remarkable insight through personal and scholarly interactions with them and their suggestions at various points during my research.

A special thanks to my family for their encouragement and valuable support. Last but not the least, I thank my beloved husband Mr. Takayuki Isogawa. Words cannot express how grateful I am to him for giving me the extra strength and motivation to accomplish my goals.

List of Publications

In this section, all the publications, along with the awards, that have resulted from the research presented as the part of this thesis are listed in the following.

Awards

- **MIRU Interactive Session Award**
  The 21st Meeting on Image Recognition and Understanding (MIRU), 2018, Japan.
- **Best Presentation Award (MVE Award)**
  IEICE Technical Committee on Media Experience and Virtual Environment (MVE), 2014, Japan.

Journal Publication

- Mariko Isogawa, Dan Mikami, Daisuke Iwai, Hideaki Kimata, and Kosuke Sato. Mask Optimization for Image Inpainting. *IEEE Access*, vol.6, pp.69728-69741 — 2018.
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Hideaki Kimata. Image quality assessment for inpainted images via learning to rank. *Multimedia Tools and Applications*, to appear — 2018.
- Mariko Isogawa, Dan Mikami, Daisuke Iwai, Kosuke Takahashi, Hideaki Kimata, and Kosuke Sato. Which is the better inpainted image? Training data generation without any manual operations. *International Journal of Computer Vision (IJCV)*, to appear — 2018.
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. Image and video completion via feature reduction and compensation, *Multimedia Tools and Applications*, vol.76, issue 7, pp. 9443-9462 — 2017.

International Conference Papers

- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Hideaki Kimata. Which is the better inpainted image? Learning without subjective annotation. *British Machine Vision Conference (BMVC)*, Selected Spotlight, 2017.
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. Eye gaze analysis and learning-to-rank to obtain the most preferred result in image inpainting. *IEEE International Conference on Image Processing (ICIP)*, pp. 3538-3542, 2016.
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. Virtual Omnidirectional Video Synthesis with Multiple Cameras for Sports

Training. *3rd International Congress on Sports Sciences Research and Technology Support (icSPORTS)*, pp. 271-275, 2015.

- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. Content Completion in Lower Dimensional Feature Space through Feature Reduction and Compensation. *14th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 156-159, 2015.
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima, "Toward Enhancing Robustness of DR System: Ranking Model for Background Inpainting", *14th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 178-179, 2015.

Symposium Papers and Talks

- Mariko Isogawa, Dan Mikami, Daisuke Iwai, Yoshinori Kusachi, and Kosuke Sato. Unnatural Region Estimation for Inpainted Image. *The 21st Meeting on Image Recognition and Understanding (MIRU)*, 2018. [MIRU Interactive Session Award]
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. Ranking model for image inpainting. *Information Processing Society of Japan, Media experience Virtual Environment (MVE)*, vol.115, no.245, p.49-54, 2015.
- Mariko Isogawa, Dan Mikami, Kosuke Takahashi, and Akira Kojima. Content completion by restoring in narrow feature space and compensating lost features. *Information Processing Society of Japan, Media experience Virtual Environment (MVE)*, vol.115, no.76, p.37-42, 2015.
- Mariko Isogawa, Dan Mikami, and Akira Kojima. Reduction and Synthesis of Feature Values for High Quality of Experience Image Completion. *Information Processing Society of Japan, Media experience Virtual Environment (MVE)*, vol.114, no.239, p.37-42, 2014. [Best Presentation Award (MVE Award)]