M.S. THESIS

Context-Driven Hybrid Image Inpainting

컨텍스트 기반 하이브리드 이미지 인페인팅

BY

CAI LU

AUGUST 2015

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Context-Driven Hybrid Image Inpainting

컨텍스트 기반 하이브리드 이미지 인페인팅

BY

CAI LU

AUGUST 2015

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Context-Driven Hybrid Image Inpainting

컨텍스트 기반 하이브리드 이미지 인페인팅

지도교수 김 태 환
이 논문을 공학석사 학위논문으로 제출함

2015년 8월

서울대학교 대학원

전기 컴퓨터 공학부

루

루의 공학석사 학위 논문을 인준함

2015년 8월

위 원 장: _____
부위원장: _____
위 원: _____

Abstract

Image inpainting, which is the filling-in of missing regions in an image, is one of the most important topics in the area of computer vision and image processing. The existing non-hybrid image inpainting techniques can be broadly classified into two types. One is the texture-based inpainting and the other is the structure-based inpainting. One critical drawback of those techniques is that their inpainting results are not effective for the images with a mixture of texture and structure features in terms of visual quality or processing time. However, the conventional hybrid inpainting algorithms, which aim at inpainting images with texture and structure features, do not effectively deal with the two items: (1) *what is the most effective application order of the constituents?* and (2) *how can we extract a minimal sub-image that may contain best candidates of inpainting source?* In this work, we propose a new hybrid inpainting algorithm to address the two tasks fully and effectively. Precisely, our algorithm attempts to solve two key ingredients: (1) (*right time*) *determining the best application order for inpainting textural and structural missing regions* and (2) (*right place*) *extracting the sub-image containing best candidates of source patches to be used to fill in a target region*. Through experiments with diverse image testcases, it is shown that our algorithm integrating the enhancements has greatly improved the inpainting quality compared to that of the previous non-hybrid inpainting methods while even spending much shorter processing time compared to the conventional hybrid inpainting methods.

keywords: image processing, hybrid image inpainting, texture and structure distribution

student number: 2013-23842

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

The task of image inpainting, which refers to filling-in one or more missing regions in an image, is one of the most important research areas in the field of computer vision and image processing. The task can be used in various applications such as image restoration, object removal, loss concealment. Guillemot and Le Meur in [1] surveyed the image inpainting methods in the literature. In particular, they observed that the pixels in some known and unknown parts of an image were very likely to share identical statistical properties or geometrical structures and reviewed the existing inpainting methods that were based on that observation. Specifically, they classified the methods into two categories: diffusion-based inpainting (e.g., [2, 3, 4, 5, 6, 7]) and exemplar-based inpainting (e.g., [8, 9, 10, 11, 12, 13]).

Diffusion-based inpainting diffuses the image information from the known region into the missing region. Bertalmio *et al.* [2] first proposed a diffusion-based inpainting method, which propagates image Laplacians from the surrounding neighborhood towards the interior of the missing region. Later, they employed Navier-Stokes equations to further improve the quality of image inpainting [3], in which they transformed the image intensity along smooth level curves into stream of fluid dynamics and the isophote lines into flow stream lines. Chan and Shen [4] proposed an inpainting framework which minimizes total variation energy of the image inside the missing region.

To reduce the processing time complexity further, Telea [5] proposed a method that estimates the image smoothness as a weighted average over known image neighborhoods and propagates the image information by using a so-called fast marching method. On the other side, Qin *et al.* [6] used anisotropic heat transfer model in which they included a texture term in their numerical implementation of the partial differential equations (PDEs) to propagate texture information. Besides the traditional PDE-based methods, it has been known that TV (total variation) regularization is able to effectively recover sharp edges in certain conditions. TV regularization regards image as a function of bounded variation and transforms image inpainting into a variational problem. The work in [7] is one of the recent TV methods. It applies the Split Bregman algorithm to TV inpainting. The strength of the diffusion-based inpainting is that the processing time is very short. However, it degrades image quality such as blurs or smoothness in texture intensive images and is not effective for images with large size missing region. For example, the introduced texture term in [6] does not effectively handle irregular textures, which means it is effective only for textures with dominant direction.

Fig. 1.1 is the inpainting results by using diffusion-based method [5]. Black regions in these two images label the inpainting regions (or the missing regions). For the *circle* image, inpainting region covers two region, white region and grey region. Inpainting result shows blur effect on the boundary line between white region and grey region, while for those inpainting parts locate totally in whiter region or grey region, the result shows no blur effect. For the *farmland* image, the inpainting region is relatively large, and covers lots of texture features like plants features. The corresponding inpainting result shows obvious blur effect. That is the general drawback of diffusion-based method which causes diffusion-based methods are not well suited for textured images, especially if the missing region is large.

On the contrary, exemplar-based inpainting is suitable for texture dominating images as shown in Fig. 1.2. It is motivated by a local region growing approach which grows the texture one pixel or one patch at a time. Wei and Levoy [8] introduced a

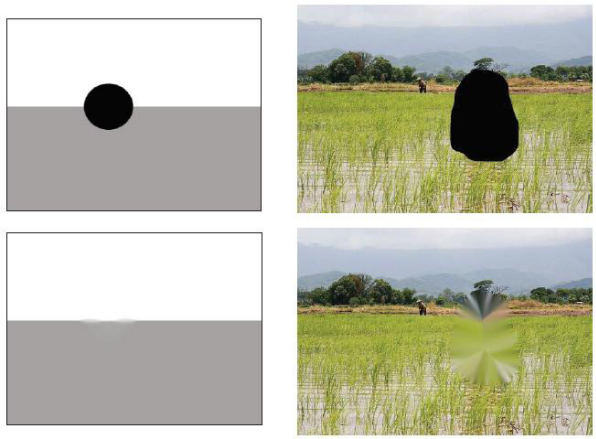

Figure 1.1: Drawbacks and benefits of diffusion-based inpainting method [5]: inpainting results of *circle* and *farmland*. Blur effects damage texture dominating regions significantly, while can be ignorable on smooth regions.

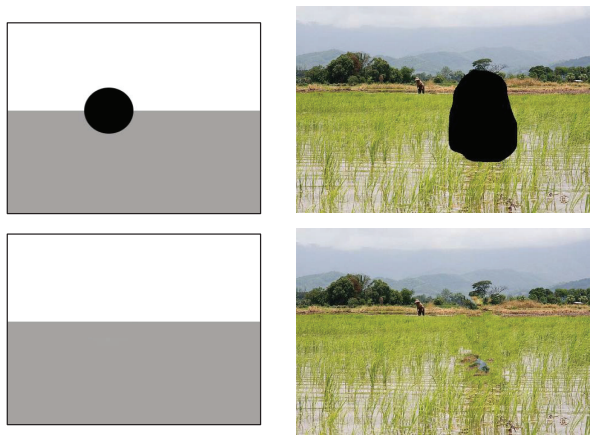

Figure 1.2: Drawbacks and benefits of exemplar-based inpainting method [9]: inpainting results of *circle* and *farmland*. Exemplar-based inpainting can preserve textures.

pixel-by-pixel exemplar-based inpainting method in which missing pixels are learned from their known neighborhood by calculating and comparing distance metrics such as SSD (sum of squared differences), and copying the central pixels of the best matches. Later, Criminisi *et al.* [9] proposed a patch-by-patch exemplar-based inpainting method in which instead of copying pixel, entire patch is copied, one at a time, to reduce run time. Since the inpainting quality highly depends on the selection of target patch to be filled at each iteration and the extraction or derivation of source patch(s) to fill-in the target patch, many works have looked for more effective methods to solve these two subproblems. One noticeable research progress on these subproblems is using the concept of 'patch sparsity' (e.g., [10, 11, 12]). The idea is that instead of copying the most similar source patch, a combination of several source patches formed by predefined waveforms (e.g., [11]) or linear combination (e.g.,[10, 12]) is used to enhance the robustness. In addition, methods (e.g., [13]) which recover the full resolution image by applying existing exemplar-based method to low resolution image as a preprocessing turned out to be effective to save run time. The main differences of the existing exemplar-based inpainting methods lie on (i) how they search for best matching patches and (ii) how they estimate the region of inpainting source. Even though lots of efforts have been made on implementing the tasks by the existing methods, we will show later that they do not fully exploit the surrounding image context of target (missing) patch for (i) and (ii).

Note that even though some methods such as [14, 15] took the texture and structure characteristics of processing units of image into account, they were effective only when image has a single or a few contiguous missing blocks. For example, [14] performed block classification for each of 8×8 missing blocks whereas our method determines the size of processing units[1] dynamically according to the images' global characteristics. In addition, [15] tried to preserve edges in the processing units rather than texture

[1] *Processing units* and *source (search) window* refer to the same meaning and used interchangeably in this work.

information while ours exactly preserves texture of image.

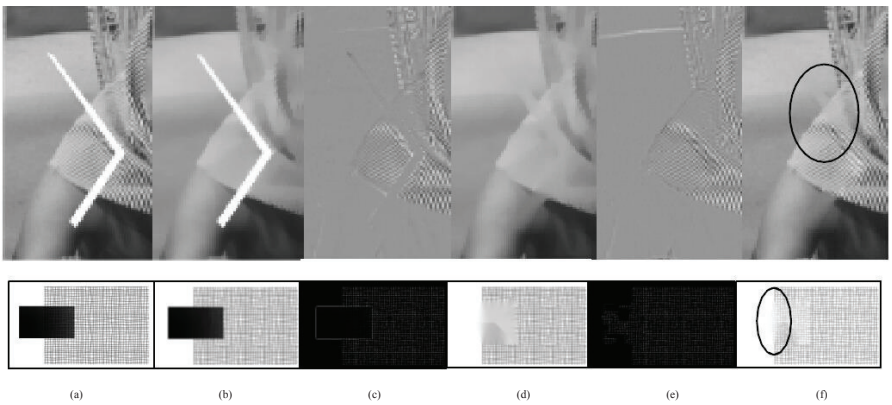

Figure 1.3: Hybrid-inpainting method step-by-step illustration of *barbala*. (a)-(g) are original image, original structure image, original texture image, inpainted structure image, inpainted texture image, inpainted result image.

To overcome the blur effect in diffusion-based inpainting and the long run time in exemplar-based inpainting, hybrid inpainting methods (e.g., [16, 17, 18]) have been proposed. Two works in [16] and [17] basically performed the following three steps: (1) separating the original image into two image layers, one containing structure information such as strong edges and corners, and the other containing texture information such as texture patterns; (2) applying an existing exemplar-based inpainting technique to the texture image layer and an existing diffusion-based inpainting technique to the structure image layer; (3) combining the two inpainted results obtained from last step into one. Fig. 1.3 takes *barbala* as an example to show the hybrid inpainting procedure. The circle region marked in (g) shows one problem of this method, the blur effect of structure image result still have effect on the final result, even though the effect is reduced by combining texture image. Another problem of this method is that instead of inpainting one image by one method, this method actually inpaints two images by two methods, which will be more time consuming and doesn't make full use of the advantages of two inpainting methods. Another kind of hybrid method, Bugeau *et al.*

[18] combined the following three basic techniques: copy-and-paste texture synthesis, geometric partial differential equations (PDEs) and coherence among neighboring pixels in one framework, and formulated it into a problem of minimizing an energy function. However, the run time is not satisfactory, taking 5 minutes for image of size 256×163.

In this work, we propose a new hybrid inpainting algorithm, called InP-h, to overcome the long processing time of the conventional hybrid methods while maintaining the visual quality. Precisely, InP-h addresses two tasks: (1) *determining the best application sequence for inpainting textual and structural missing target patches* and (2) *extracting the sub-image (i.e., source window) containing the best candidates of source patches*.

The rest of the paper is organized as follows. Section 2 reviews a state-of-the-art exemplar-based inpainting method, which is integrated into InP-h, together with our improvement on critical limitation (i.e., task 2) of exemplar-based inpainting. Section 3 describes our enhanced execution procedure (i.e., task 1) and details of our hybrid inpainting algorithm integrated with exemplar-based and diffusion-based inpaintings. Then, comparisons of our inpainting results with those produced by diverse existing methods are shown in Section 4. Finally, a conclusion of the work is given in Section 5.

Chapter 2

Exemplar-based Inpainting: Review and Enhancement

This section overviews one of the most up-to-date methods of exemplar-based inpainting [12] (Subsection 2.1), which we adopt for our integration, followed by proposing an improvement on a core part of exemplar-based inpainting (Subsection 2.2).

2.1 Preliminary: A State-of-the-Art Exemplar-based Inpainting

Let I and Ω denote an input image and a sub-region to be inpainted, respectively. Then, $I - \Omega$ (= Φ) represents the source region of image. The basic idea of exemplar-based inpainting is to utilize the image information in Φ to recover the inpainting region iteratively one patch at a time. The most recent method of exemplar-based inpainting in [12] consolidated the following two procedures:

1. *Processing order of patches*: Fig. 2.1 shows the diagram to explain the steps of exemplar-based inpainting. It paints the patches in the missing region, one patch at a time, according to the patches' priority. For example, q_1, q_2, q_3, $\cdots$ in Fig. 2.1 are the source pixel points, p is a point on the boundary of missing region, and Ψp and Ψq_i are the patches centered at points p and q_i, respectively.

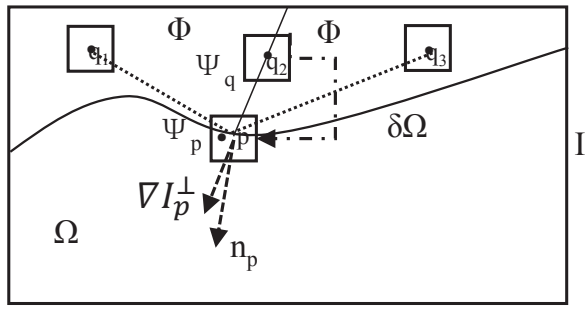

Figure 2.1: Diagram to explain the procedure of exemplar-based inpainting. A target patch Ψp is selected at each iteration based on the priority decided by several factors such as the confidence value which is the number of source pixels in a certain patch, the data value which represents the directions and values of $\bigtriangledown I_p^{\perp}$ and n_p, followed by selecting a best source patch Ψq_i or a set of patches to fill the target patch Ψp.

$\bigtriangledown I_p^{\perp}$ along the direction of the existing edge in the source region represents the isophote, where direction is represented by arrow and intensity by the length of arrow at point p, and n_p is the normal to the boundary $\delta\Omega$ at point p. After comparing the target patch Ψp with every candidate source patch (e.g., Ψq_1, Ψq_2, Ψq_3), if Ψq_i is judged as the best match to the Ψp, Ψq_i is copied to Ψp. The selection of target patch to be filled in the current iteration is determined based on several criterias. One representative computation of the priority is shown in *Eqs.*(2.1), (2.2), and (2.3) according to [10]:

$$P\left(p\right) = C\left(p\right) \times D\left(p\right), \tag{2.1}$$

$$C\left(p\right) = \frac{\sum_{q \in \Psi p \cap (I-\Omega)} C(q)}{|\Psi p|}, \tag{2.2}$$

$$D\left(p\right) = \frac{|\nabla I_p^{\perp} \bullet n_p|}{\alpha} \tag{2.3}$$

where $P(p)$ indicates the priority value of the patch Ψp. $C(p)$ and $D(p)$ represent the number of known pixels in Ψp and the strength of isophotes hitting the boundary $\delta\Omega$, respectively. α is set to 255 to scale $D(p) \in [0, 255]$.

Guillemot, Turkan, Le Meur, and Ebdelli [12] updated the processing order defined in *Eq.*(2.1) to take into account the textures which are not perpendicular to the front line. They introduced a new term $E(p)$ in *Eq.*(2.4) to represent the edgeness of patch Ψ_p and included it to $P(p)$ in *Eq.*(2.5):

$$E\left(p\right) = \frac{\sum_{q \in \Psi_p \cap \Phi} \gamma(q \in Edge)}{|\Psi_p \cap \Phi|}, \tag{2.4}$$

$$P\left(p\right) = C\left(p\right) \times D\left(p\right) \times E\left(p\right), \forall p \in \delta\Omega \tag{2.5}$$

where *Edge* set of pixels is determined by Canny edge detector and $\gamma(\cdot)$ is a binary function which returns 1 if the input is true and 0 otherwise.

2. *Searching for the most similar source patches*: The SSD (sum of squared differences) metric has been widely used to search for patches similar to the target patch. It measures the difference and cross-correlation between the color values of pixels. However, it is shown that using SSD tends to copy pixels from uniform regions [1]. To improve the accuracy of the measurement, the combined similarity (distance) metric of the SSD and weighted Bhattacharya has been proposed [1, 13]:

$$d_{SSD,BC}(\Psi_p, \Psi_{q_i}) = d_{SSD}(\Psi_p, \Psi_{q_1}) \times (1 + d_{BC}(\Psi_p, \Psi_{q_1})) \tag{2.6}$$

where Ψ_p and Ψ_{q_i} are the target patch and candidate source patch, respectively, and d_{SSD} and d_{BC} indicate the SSD metric and weighted Bhattacharya metric, respectively.

The most time consuming part of exemplar-based inpaintings is finding the top K most similar patches of Ψ_{q_i}, which is called K nearest neighbor (K-NN) searching. This means that identifying a minimal searching window is critical to reduce the time for K-NN searching since as the searching window increases, the number of candidate source patches to be checked increases. The current scheme sets the size of searching window by a pre-defined radius (e.g., [9, 12]). Once the top K source patches are extracted, the K patches are combined to estimate the target patch by applying a neighbor embedding (NE) technique. The locally linear embedding technique with low-dimensional neighborhood representation (LLE-LDNR) [12] has been proven to be one of the most effective NE techniques. We adopt LLE-LDNR in our hybrid inpainting framework.

2.2 Context-Driven Determination of Window Sizes

As mentioned before, searching window size affects the run time significantly, and most up-to-date exemplar-based methods use pre-defined radius value to restrict searching window size. In order to save the run time, the searching window size should be as small as possible. However, as shown in Fig. 2.2, images with different sizes and different target regions achieve the best results at different searching window size. 40 for *bungee*, 80 for *dog*, 120 for *sea*. Thus an automatical way to decide the searching window size is needed.

Our determination of searching window sizes for texture-intensive target patches is based on the following intuition:

1. A block (B_i) with size of $n\text{-}pixel \times n\text{-}pixel$ to be included in the searching window is likely to be placed close to the target patch block (Ψ).

2. Inclusion of a block with more texture feature into searching window is likely to lead a better inpainting quality.

Figure 2.2: Exemplar-based inpainting [12] results of images *bungee*, *dog*, and *sea* with different searching window sizes.

Based on the intuition, we expand the searching window starting from the target basic block by iteratively checking if it is beneficial to include the basic block on the boundary into the searching window. The guideline for the inclusion of a block B_i is based on the value of $\rho(B_i)$, which measures the *degree of textureness* of that block:

$$\rho(i) = \beta_1 \times \frac{\sum_{q \in B_i \cap \Phi} \gamma(q \in I_{s\_edge}(i))}{|B_i \cap \Phi|} + \beta_2 \times \frac{\sum_{q \in B_i \cap \Phi} \gamma(q \in I_{w\_edge}(i))}{|B_i \cap \Phi|} \quad (2.7)$$

where B_i is a block of size $n\text{-}pixel \times n\text{-}pixel$ and should have at least one known pixel, and $\rho(i) = -1$, otherwise. (B_i with $\rho(i) = -1$ means that all the pixels in B_i are missing (i.e., unknown).) I_{s_edge} and I_{w_edge} represent the sets of pixels on the 'strong' and 'weak' edges in the *Edge* set produced by the application of Canny edge detector to B_i, respectively. Thus, the first and second terms indicate the portions of the known pixels of B_i which exhibit strong and weak edges, respectively. β_1 and β_2 are weighting factors. The searching window will be expanded as long as the window covers the blocks whose ρ values are close to the ρ value of the target block. The details of setting the parameter values and procedure of window expansion will be discussed in Subsection 3.3.

Chapter 3

The Proposed Context-Driven Hybrid Inpainting

3.1 Overall Flow

Fig. 3.1 depicts the flow diagram of our proposed hybrid inpainting algorithm InP-h. InP-h consists of three steps: (Step 1) a pre-processing is performed to extract texture feature related data from input image I_0; (Step 2) the texture-intensive target patches are filled in sequentially by applying an exemplar-based inpainting technique; (Step 3) the remaining target patches are then filled all together by applying a diffusion-based inpainting technique.

Application order for inpainting: a hybrid inpainting can consider several options for the application order of exemplar-based inpainting and diffusion-based inpainting. Three options are feasible: (1) *diffusion-first*, (2) *exemplar-first*, and (3) *diffusion-exemplar-alternate*. From the facts that our objective is to reduce the processing time of hybrid inpainting without losing the inpainting quality and most diffusion-based inpainting techniques can formulate multiple target patches to be inpainted *all together* into a set of partial differential equations, the first and second options would be more acceptable than the last option in terms of saving processing time. Furthermore, since contrary to exemplar-based inpainting, diffusion-based inpainting requires the processing time for selecting target patches in the execution order of the first option, the second

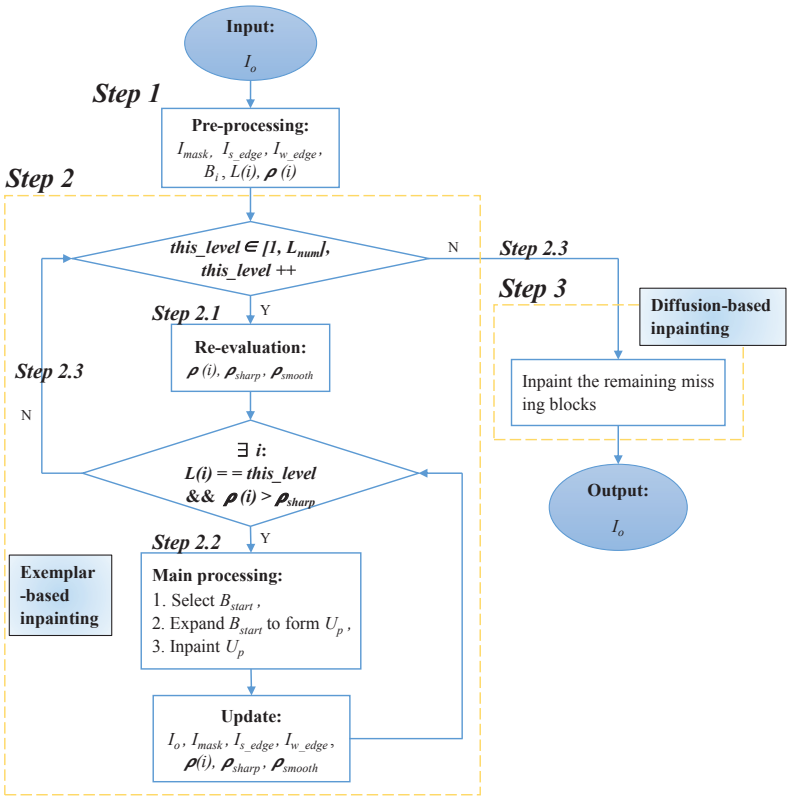

Figure 3.1: Algorithmic flow of our proposed hybrid inpainting method InP-h. It consists of three steps, which are explained in Subsections 3.2, 3.3, and 3.4. Note that comparing to the pure exemplar-based inpainting, introducing diffusion-based inpainting speeds up the processing time while the adaptive window sizing in Step 2 speeds up the processing time of exemplar-based inpainting phase without degrading image quality.

option will save more time than the first one. Note that InP-h reduces the processing time spent in the exemplar-based inpainting (Step 2) by our window resizing scheme that is driven by the context of the surrounding image of the target patch, which will be discussed in Subsection 3.3.

3.2 Step 1: Pre-processing

- *Generation of auxiliary images*: From I_0, InP-h extracts three types of images: I_{mask} (mask image), I_{s_edge} (strong edges), and I_{w_edge} (weak edges).

  - I_{mask}: it is used to mark the region to be inpainted, i.e., setting the pixels in that region to white and the rest to black.
  - I_{s_edge}: the Canny edge detector assigns a value for each pixel to indicate the degree of edgeness in I_0. I_{s_edge} is the collection of pixels whose edgeness values are greater than 0.9. I_{s_edge} represents strong textures.
  - I_{w_edge}: similar to the definition of I_{s_edge}, I_{w_edge} is the collection of pixels whose edgeness values are in between 0.3 and 0.9. I_{w_edge} represents weak textures.

- *Initialization of parameters*: Each of I_0, I_{mask}, I_{s_edge}, and I_{w_edge} is then uniformly partitioned into basic blocks B_i of $n \times n$ size. For each B_i, two parameters $\rho(i)$ and $L(i)$ are initialized.

  - $\rho(i)$: it is computed by the formulation in *Eq.*(2.7). A high value of $\rho(i)$ implies that the image in block B_i is skewed toward texture-intensive while a low value means the image is skewed to structure-intensive. The $\rho(\cdot)$ values will be used to make a decision on the inclusion of the corresponding blocks into the searching window.
  - $L(i)$: it is the distance between B_i and Φ, from the location of B_i to the closest block on the boundary of the missing region that contains B_i. For

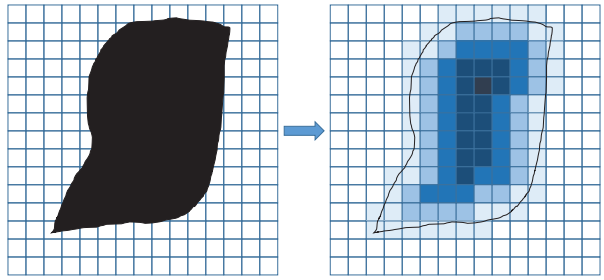

Figure 3.2: The distribution of target basic block level $L(i)$. The left image shows the source region and target region, the right image shows the $L(i)$. Both the black region in the left image and the curve in the right image represent the inpainting region, and in the right image, the value of $L(i)$ increases with the color depth.

B_i with no unknown pixel, $L(i) = 0$ and for B_i with mixture of known and unknown pixels, $L(i) = 1$, and so on. Fig. 3.2(b) shows the distribution of $L(\cdot)$ values for the input image I_0 where the source and missing regions are marked with white and black colors, respectively. A high (low) value of $L(i)$ implies that block B_i is likely to have a low (high) chance to be selected as a target patch. (Note that the target patch selection is determined based on the values of $L(\cdot)$ and $\rho(\cdot)$, as indicated in the second condition test in Fig. 3.1. The details will be explained in Subsection 3.3.)

3.3 Step 2: Exemplar-based Inpainting

Step 2 consists of three sub-steps: (Step 2.1) re-evaluation of parameters, (Step 2.2) selection of target patch, generation of source window and inpainting, and updating parameters, and (Step 2.3) termination conditions.

- *(Step 2.1) Re-evaluation of parameters*: Just before this step, our algorithmic flow asserts that $\rho(\cdot)$ is 0 for every block with no known pixel at the current iteration. This step estimates the $\rho(\cdot)$ values for such blocks by referring the $\rho(\cdot)$

values of their neighbor blocks. The $\rho(\cdot)$ is computed by

$$\rho(i) = \begin{cases} \rho(i) & \text{if } \rho(i) \geq 0, \\ \eta \times \max\limits_{B_j}(\rho(j)) & \text{if } \rho(i) = -1, \end{cases} \tag{3.1}$$

in which B_j represents every neighbor of B_i such that $L(j) = L(i) - 1$ and η is a control parameter in [0, 1].

From the $\rho(\cdot)$ values of all blocks, the average (ρ_{avg}), minimum (ρ_{min}), and maximum (ρ_{max}) are computed. Then, we define two parameters ρ_{smooth} and ρ_{sharp}:

$$\rho_{smooth} = \rho_{min} + \epsilon_1 \cdot (\rho_{avg} - \rho_{min}), \tag{3.2}$$

$$\rho_{sharp} = \rho_{avg} + \epsilon_2 \cdot (\rho_{max} - \rho_{avg}) \tag{3.3}$$

in which ϵ_1 and ϵ_2 are in $[0, 1]$, and empirically set to 0.5 and 0 in InP-h, respectively.

By using the values of ρ_{smooth} and ρ_{sharp}, we divide the interval $[\rho_{min}, \rho_{max}]$ into three sub-intervals: $rg_{small} = [\rho_{min}, \rho_{smooth}]$, $rg_{middle} = [\rho_{smooth}, \rho_{sharp}]$, and $rg_{large} = [\rho_{sharp}, \rho_{max}]$.

- *(Step 2.2) Selection of target patch and generation of source window*: We select, among all target patches (basic blocks) whose $L(\cdot)$ values are the level number (*this_level*) in the current iteration of the upper-loop in Fig. 3.1, the one with the largest $\rho(\cdot) \in rg_{large}$. For example, Fig. 3.3(a) shows the classification of blocks into rg_{smooth} (red), rg_{mix} (yellow), and $rg_{texture}$ (blue). (The missing region is marked with gray color.) Let the $L(\cdot)$ values of B_1 through B_6 all equal to the value of *this_level*, and $\rho(6) \in rg_{largest}$ is the largest. Then, B_6 will be selected as the starting target block B_{start} for expansion.

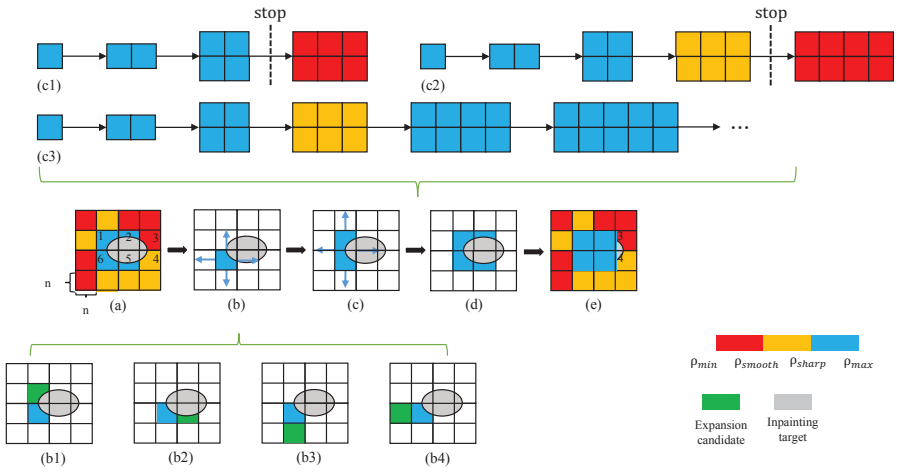

Figure 3.3: An example illustrating the selection of starting target patch (basic block), B_{start}, and generation of source window by expanding the target patch: (a)∼(e) steps of window expansion, (b1)∼(b4) candidate expansion directions, (c1)∼(c3) termination of window expansion.

InP-h iteratively expands B_{start} toward one of four directions: left, right, up, and down. (See Fig. 3.3(b).) The expansion direction is chosen based on the ρ value of the resulting window of each expansion. Let ρ_{exp} be the largest value and the corresponding direction is chosen for expansion. The process of iterative expansion considers the following two cases.

Case-1 (Figs. 3.3(c1) and (c2)): the expansion stops if $\rho_{exp} \in rg_{small}$ since the expansion leads to dimming texture feature.

Case-2 (Fig. 3.3(c3)): the expansion continues as long as $\rho_{exp} \in rg_{middle} \cup rg_{large}$.

In addition, there are three exceptions for handling the termination/continuation of the expansion other than Cases 1 and 2.

Exp-1 : If the expansion stops too early, the resulting source window is so small that the source information is not enough to recover the target block natu-

rally. Thus, we continue the expansion if the window size is below a lower threshold.

Exp-2 : A long expansion to one direction may cause unbalanced information of source image to be used for painting target block. Thus, a minimal control for expansion direction is internally installed in InP-h.

Exp-3 : Too large window produced by expansion may contain unnecessary or redundant texture information with respect to the target block. Thus, we control the number of expansion iterations so that the resultant window is below an upper threshold.

Once a source window is extracted by the expansion, a conventional exemplar-based inpainting is applied to the target block[1] in the window using the source information in the window. Then, images I_0, I_{mask}, I_{s_edge}, I_{w_edge}, and ρ values are updated accordingly, as illustrated in Fig. 3.3(e). Then, the lower-loop in Fig. 3.1 (i.e., Step 2.2) repeats until there is no block which has larger ρ value than ρ_{sharp} or has L value of *this_level*.

- *(Step 2.3) Termination*: The iteration of exemplar-based inpainting stops when there is no block to be inpainted, which means there is no block whose ρ value is no less than ρ_{sharp}. This condition is checked at the two loops in Fig. 3.1.

3.4 Step 3: Diffusion-based Inpainting

The target blocks that remain after Step 2 will be those located on the relatively smooth regions. Diffusion-based inpainting is applied to all the remaining blocks in Ω all together by generating a set of partial differential equations for the blocks. Note that since we estimate the value of each target pixel with the average of the values of its neighbor pixels, the window extraction task is no longer needed in this step.

[1]Note that the block could have multiple basic blocks with unknown pixels.

Chapter 4

Experimental Results

We have tested our proposed hybrid method InP-h on a set of images in the literature, and compared against several state-of-the-art inpainting methods. For exemplar-based inpainting, InP-h internally used the modified version of [12] described in Section 2 in which our context-driven scheme for extracting source window (sub-image) is installed while for diffusion-based inpainting the method in [5] is installed. The existing inpainting methods whose results are compared with ours are that in [12] for pure exemplar-based inpainting, that in [5] for pure diffusion-based inpainting, that in [17] for hybrid inpainting, the super-resolution-based inpainting in [13], and the total variation (TV) inpainting in [7].

All experiments were run on a 2.93-GHZ Intel i7 with 4.00 GB RAM and performed in three folds: (1) checking the effectiveness of generating source windows, (2) checking the effectiveness of performing object removal, and (3) checking the effectiveness of performing loss concealment.

- *Assessing the performance of adaptive window sizing*: At each iteration in Step 2 of InP-h, a source window U_i containing a target block for inpainting is extracted by examining the blocks surrounding the target block. Fig. 4.1 shows extracted windows for images *circle*, *bungee*, and *camera*. It is visually confirmed that the size of windows is well controlled and sufficiently enough to enable nat-

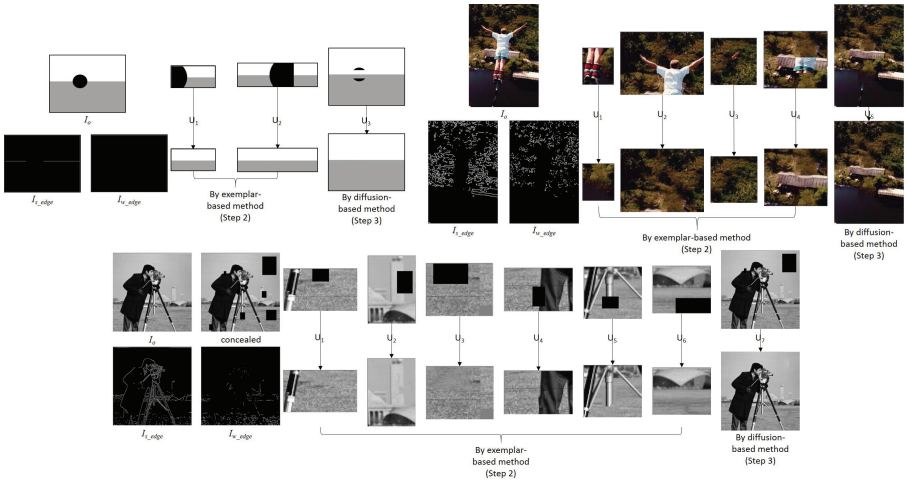

Figure 4.1: InP-h inpainting procedure for images *circle*, *bungee*, and *camera*. Source windows U_1, U_2, $\cdots$ are generated for inpainting. The images in the windows before and after the applications of exemplar-based inpainting in Step 2 of InP-h and diffusion-based inpainting in Step 3 of InP-h are shown.

ural inpainting.

- *Assessing the effectiveness of object removal*: Fig. 4.2 shows comparison of visual quality. It is seen that diffusion-based inpainting produces blur effect. Even though the total variation (TV) method reduces the blur effect, it still fails to recover the texture information, in particular, for image *bungee*. As we mentioned, the size of source window greatly affects the inpainting quality in exemplar-based inpainting. Either too large or too small may produce inferior quality. Here, we choose two (fixed) searching window sizes: $radius = \infty$ to consider the whole image as searching source and $radius = 80$ to consider the circled image centered at target block with radius of 80 pixels. For image *baseball*, matching the target block with the blocks in the whole image failed. For image *sea*, matching with window of $radius = 80$ failed as well. The hybrid method tested here used the pure exemplar-based method of $radius = 80$ [12] and the

pure diffusion-based method [5]. It is seen that the blur effect in the structure layer poses a strong negative effect on the final result and the recovered texture is almost invisible. In addition, the super-resolution-based method shows an unstable performance.

Figure 4.2: Comparison of visual quality of object removal for images *bungee*, *dog*, *circle*, *elephant*, *baseball*, and *sea*, in sequence from left to right and top to bottom. The tested methods are labelled under the images.

Table 4.1 summarizes the run times used by the conventional inpainting methods and our InP-h for inpainting the images in Figs. 4.2 and 4.3. When the

searching window is set to the whole image, the exemplar-based method takes considerably long processing time. The run time specified in Table 4.1 used by the super-resolution-based method in [13] excludes the time spent for SR phase (super-resolution phase means re-generating the high resolution image from the low resolution image). Note that even though the diffusion method uses much short run time, the blur effect is too severe to be ignored.

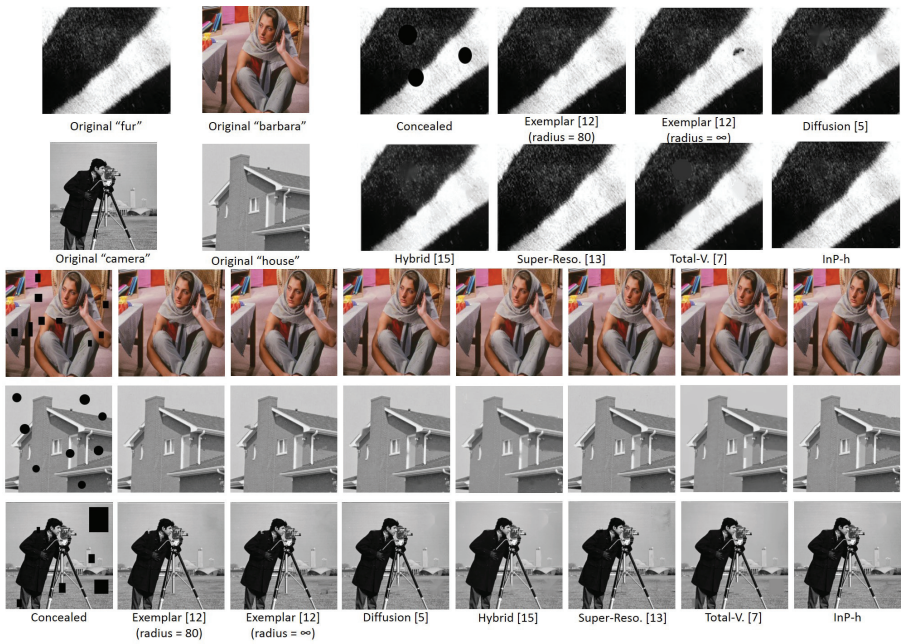

Figure 4.3: Comparison of the visual quality of inpainting results for loss concealment on images *fur*, *barbara*, *house*, and *camera*. The tested inpainting methods are listed below the corresponding images.

- *Assessing the effectiveness of loss concealment*: Fig. 4.3 summarizes a comparison of visual quality. All concealments were selectively chosen so that they can cover both texture region and smooth region uniformly. The difference between the visual qualities of the results is measured by using the PSNR metric, as listed in Table 4.2. Since the diffusion-based inpainting recovers the target region by

Table 4.1: Comparison of run time (sec.) spent by the existing inpainting methods and ours

	bungee	*circle*	*dog*	*elephant*	*baseball*
Missing areas	13.8%	4.9%	7.7%	17.3%	13.1%
Exemplar [12] ($radius = 80$)	11.66	9.31	6.56	19.38	17.07
Exemplar [12] ($radius = \infty$)	32.89	79.72	22.73	220.7	177.1
Diffusion [5]	0.06	0.06	0.04	0.16	0.13
Hybrid [17]	12.41	11.51	7.08	20.4	17.76
Super-Reso. [13]	45	36	24	137	96
Total-Variation [7]	11	9	2	30	28
InP-h	**6.59**	**1.4**	**1.22**	**5.17**	**12.46**
	sea	*barbara*	*fur*	*camera*	*house*
Missing areas	10.9%	4.5%	6.3%	8.6%	7.1%
Exemplar [12] ($radius = 80$)	24.47	18.64	4.47	22.9	5.4
Exemplar [12] ($radius = \infty$)	505.8	221.4	10.03	376.7	18.7
Diffusion [5]	0.2	0.1	0.02	0.15	0.03
Hybrid [17]	25.76	19.45	4.67	23.6	5.8
Super-Reso. [13]	192	74	21	203	41
Total-Variation [7]	50	51	1	16	4
InP-h	**8.45**	**3.13**	**1.58**	**5.39**	**1.22**

using the neighbor pixels' average value, it achieves relatively stable PSNR results. However, when taking the blur effect as well as the numbers in the Table 4.2 into consideration, our InP-h outperforms the diffusion-based inpainting. Finally, as shown in Table 4.1, InP-h exhibits a better performance than the other methods in run time. Note that image *circle* in the experiments was tested as both cases of object removal, removing the dark circle and loss concealment, recovering information under the dark circle.

Table 4.2: Comparison of the values of PSNR metric for the images inpainted by the existing methods and ours

	barbara	*fur*	*camera*	*house*	*circle*
Exemplar [12] ($radius = 80$)	34.8	33.4	34.5	38.9	31.0
Exemplar [12] ($radius = \infty$)	35.2	27.8	33.8	35.6	31.1
Diffusion [5]	35.7	31.9	37.8	34.8	42.0
Hybrid [17]	27.5	27.5	31.2	31.6	41.2
Super-Reso. [13]	32.1	29.0	33.5	37.6	36.8
Total-Variation [7]	34.9	34.8	35.9	35.8	41.6
InP-h	**35.8**	**34.8**	**37.0**	**38.1**	**64.7**

Chapter 5

Conclusion

A comprehensive image inpainting method was proposed to enhance the two critical tasks in the prior hybrid methods, which are (1) *setting up the best application order for inpainting textural and structural missing regions* and (2) *extracting the sub-image containing best candidates of source patches* to be used to fill in a missing region. By integrating our *execution-order analysis based solution* to task 1 and our image *context-driven source image extraction solution* to task 2, we were able to consistently improve inpainting quality compared to that of the previous non-hybrid inpainting methods while even spending much shorter processing time compared to the conventional hybrid inpainting methods.

Bibliography

[1] C. Guillemot and O. Le Meur, "Image inpainting: overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127-144, Jan. 2014.

[2] M. Bertalmio and G. Sapiro and V. Caselles and C. Ballester, "Image inpainting," in *ACM SIGGRAPH*, New Orleans, USA, July 2000. pp. 417-424.

[3] M. Bertalmio and A. L. Bertozzi and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI, USA, Dec. 2001. pp. 355-362.

[4] T. F. Chan and J. Shen, "Local inpainting models and TV inpainting," *SIAM Journal of Applied Mathematic*, vol. 62, no. 3, pp. 1019-1043, Mar. 2001.

[5] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 25-36, 2004.

[6] C. Qin and S. Wang and X. Zhang, "Simultaneous inpainting for image structure and texture using anisotropic heat transfer model," *Multimedia Tools and Applications*, vol. 56, no. 3, pp. 469-483, Sep. 2012.

[7] P. Getreuer, "Total variation inpainting using split bregman," *Image Processing On Line*, vol. 2, pp. 147-157, July 2012.

[8] L-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *ACM SIGGRAPH*, New Orleans, USA, July 2000. pp. 479-488.

[9] A. Criminisi and P. Perez and K. Toyama, “Region filling and object removal by exemplar-based inpainting,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[10] A. Wong and J. Orchard, “A nonlocal-means approach to exemplar-based inpainting,” *IEEE Transactions on Image Processing*, pp. 2600-2603, Oct. 2008.

[11] Z. Xu and J. Sun, “Image inpainting by patch propagation using patch sparsity,” *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1153-1165, May 2010.

[12] C. Guillemot and M. Turkan and O. Le Meur and M. Ebdelli, “Object removal and loss concealment using neighbor embedding methods,” *Signal Processing: Image Communication*, vol. 28, no. 10, pp. 1405-1419, Nov. 2013.

[13] O. Le Meur and M. Ebdelli and C. Guillemot, “Hierarchical super-resolution-based inpainting,” *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3779-3790, Oct. 2013.

[14] S. D. Rane and G. Sapiro and M. Bertalmio, “Structure and texture filling-in of missing image blocks in wireless transmission and compression applications,” *IEEE Transactions on Image Processing*, vol. 12, no. 3, pp. 296-303, Mar. 2003.

[15] C. Qin and F. Cao and X. P. Zhang, “Efficient image inpainting using adaptive edge-preserving propagation,” *The Image Science Journal*, vol. 59, no. 4, pp. 211-218, Aug. 2011.

[16] M. Bertalmio and L. Vese and G. Sapiro and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882-889, Aug. 2003.

[17] J. Wu and Q. Ruan, “A novel hybrid image inpainting model,” in *International Conference on Audio, Language and Image Processing (ICALIP)*, Shanghai, China, July 2008. pp. 138-142.

[18] A. Bugeau and M. Bertalmio and V. Caselles and G. Sapiro, “A comprehensive framework for image inpainting,” *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2634-2645, Oct. 2010.

[19] L. Liang and C. Liu and Y. Xu and B. Guo and H. Shum, “Real-time texture synthesis by patch-based sampling,” *ACM Trans. Graph*, vol. 20, no. 3, pp. 127-150, Oct. 2001.

[20] J. Stark and M. Elad and D. Donoho, “Image decomposition via the combination of sparse representations and variational approach,” *IEEE Trans. Image Processing*, vol. 14, pp. 1570-1582, Oct. 2005.

초 록

이미지 중에 손상된 부분을 채우고 원하지 않은 대상을 치우는 것은 이미지 인페인팅이라고 한다. 이 것은 컴퓨터 비전이나 영상 처리 분야에서 중요한 주제로 알려졌다. 기존 이미지 인페인팅 기술은 크게 텍스처 기반 인페인팅과 구조 기반 인페인팅 두 유형으로 분류될 수 있다. 이름에서 알 수 있듯 각각 텍스처 정보나 구조 정보를 이용해서 인페인팅하므로, 이로 인한 장점과 단점이 있다. 텍스처와 구조 특정을 동시에 가지고 있는 이미지의 경우, 둘 중 하나를 이용하는 기술 보다 둘을 모두 고려하여 활용하는 하이보리드 기술이 더 좋은 결과를 얻을 수 있다. 흔히 알려진 하이브리드 기술이 두 가지 문제를 효율적으로 해결하지 못 한다. (1) 텍스처 기반 인페인팅과 구조 기반 인페인팅 방법을 어떤 순서로 사용해야 하느가? (2) 어떻게 인페인팅할 이미지 후보들을 포함하는 최소의 서브이미지를 추출할 수 있는가?

이 논문에서 이 두 가지 문제점을 해결하고 적절한 시점에 적절한 위치에 있는 정보를 이용해 더 나은 인페인팅 기술을 적용하는 새로운 하이브리드 방법을 제시했다. 다양한 이미지를 테스트 한 결과, 제안된 알고리즘은 텍스처 기반 인페인팅과 구조 기반 인페인팅 방법의 장점을 유지하면서 기존 방법보다 더 빠른 기간 내에 더욱 안정된 이미지 품질을 보였다.

주요어: 영상 처리, 하이보리드 이미지 인페인팅, 텍스처 구조 특정 분포

학번: 2013-23842