Adaptive Depth Guided Image Completion for Structure and Texture Synthesis

by

Michael Luigi Ciotta

Bachelor of Engineering (B.Eng), McMaster University, 2013

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Adaptive Depth Guided Image Completion for Structure and Texture Synthesis

Master of Applied Science 2015

Michael Luigi Ciotta

Electrical and Computer Engineering

Ryerson University

Abstract

The problem of synthesis of missing image parts represents an interesting and challenging area of image processing and computer vision with significant potential. This thesis, focuses on an adaptive depth-guided image completion method that addresses the image completion problem using information contained in the rest of the image. The completion process is separated into structure and texture synthesis. A method is first introduced for completing the respective depth map through the use of a diffusion-based operation, preserving global image structure within the unknown region. Building upon the state of the art exemplar based inpainting technique of Barnes et al., we complete the target (unknown) region by matching to and blending source patches drawn from the rest of the image, using the reconstructed depth information to guide the completion process. Secondly, for each target patch, we formulate an adaptive patch size determination as an optimization problem that minimizes an objective function involving local image gradient magnitude and orientations. An extension to the coherence-based objective function introduced by Wexler et al. is then introduced, which not only encourages coherence of the respective target region with respect to the source region in colour but also in depth. We further consider the variance between patches in the SSD criteria for preventing error accumulation and propagation. Experimental results show that our method can provide a significant improvement to patch-based image completion algorithms shown by PSNR and SSIM calculations as well as a qualitative subjective study.

Acknowledgements

First and foremost, I would like to thank my parents, for their love and support throughout my life. Thank you both for encouraging me and giving me the strength to reach for the stars and chase my dreams. My little sisters and brother deserve my wholehearted thanks as well, since you tirelessly listened to my ideas and offered words of encouragement when it was most needed. Without you in my life this thesis would never have been made possible.

I would like to thank my supervisor, Prof. Dimitri Androutsos, for his guidance, expert advice, unfailing patience and especially for his confidence in me. I would also like to thank Dr. Richard Rzeszutek for taking his time and providing endless feedback for my thesis.

Finally, to all my friends, thank you for your words of encouragement and for believing in me, your friendship makes my life a wonderful experience.

Dedication

I dedicate my thesis to my beloved grandmother, Alda Picone, who unfortunately didn't stay in this world long enough to see her grandson become an engineer. May you continue to rest in peace, as you will be in my heart forever.

Contents

List of Tables

List of Figures

List of Symbols

α - Exponential factor

ϵ - Variance Membership threshold value

Ω - Target/Mask Region - Area of the image that needs to be replaced or repaired

$\partial\Omega$ - Boundary - Boundary between the source and target region

ϕ - Controls the variance membership transition sharpness

Ψ - Source/Allowed Region - Area of the image that serves to fill the missing part of the image

ψ_p - Target Patch - square, centrally symmetric square neighborhood of a pixel p with this pixel placed in the center

ψ_q - Exemplar selection patch/Known Patch - square, centrally symmetric square neighborhood of a pixel q with this pixel placed in the center

c^i - Colour value of pixel p

$d(\psi_p, \psi_q)$ - Distance metric - Corresponding distance metric between ψ_p and ψ_q

$dist(p, T)$ - spatial distance from p to the boundary of the target region T

i - Iteration

$N_8(p)$ - 8-Connected neighbourhood of pixel p

q_R - Randomly selected exemplar

R_i - random value between [1,-1]

W - Search Window

w^i - per-pixel weighting

Optimal Exemplar - Patch with minimal distance (computed using distance metric)

Patch Size - size of the side of the patch; patch size is usually an odd number since pixel under consideration is in the center of the patch

Chapter 1

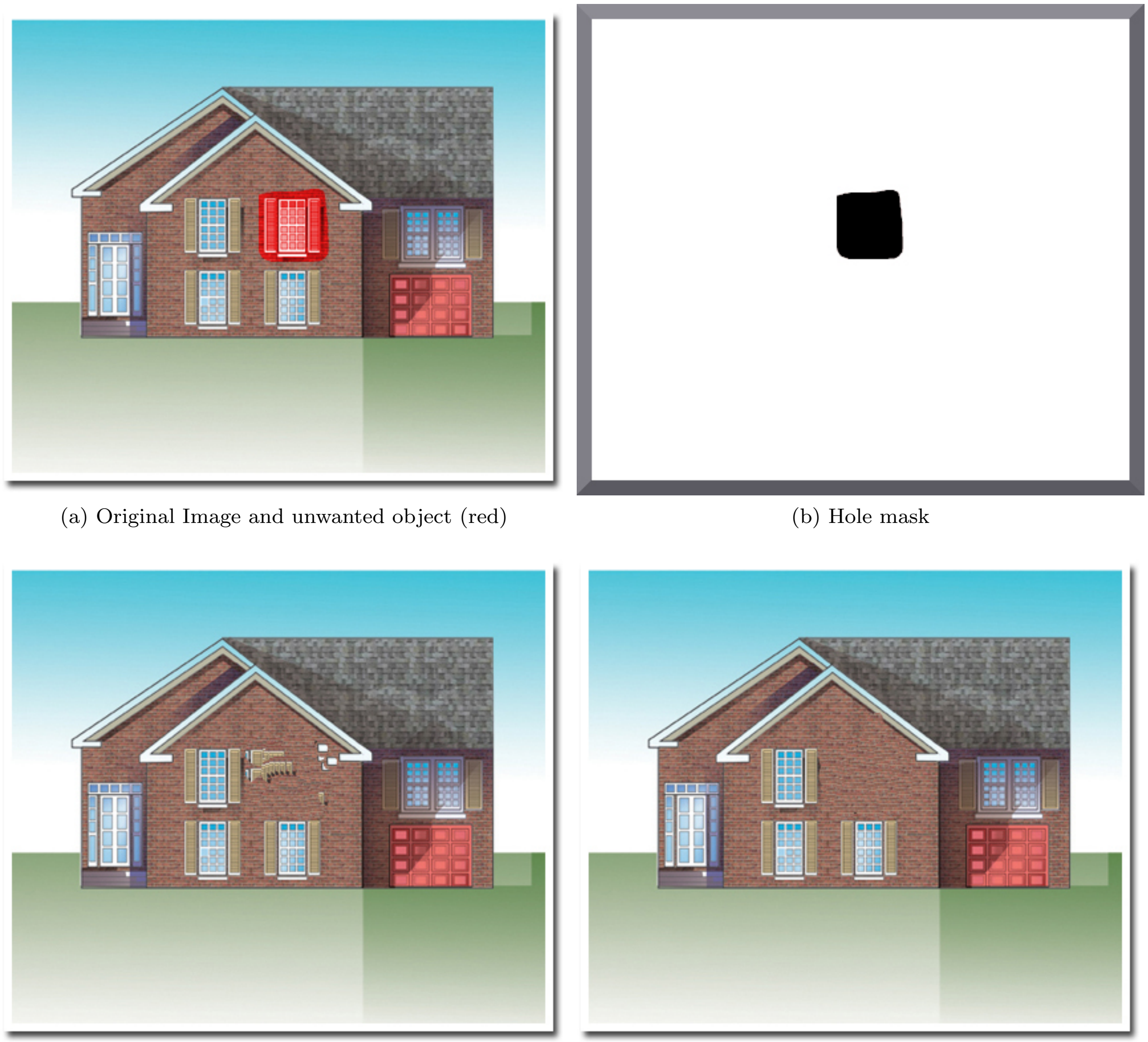
Introduction

1.1 What is Image Completion?

Image completion[1] is the process of filling in parts of an image that are considered missing, either due to damage or removal of unwanted elements. It can be considered part of a wider field of artistic restoration, the roots of which date back to the Renaissance era[2]. Modern image processing techniques allow us analyze and manipulate data of a digitized image in order to improve its overall quality. Our goal is to perform this artistic restoration on digital images restoring missing parts of an image through algorithm design on a computer. Fixing these flaws correctly is important in many applications such as removing scratches and stains from deteriorated digital images, recovering lost data due to image acquisition, compression and transmission, removal of unwanted objects/markings in photos, etc [2–7]. While the human eyes can often see the visually plausible solution of the missing image information based upon the Gestalt principle [8], finding an algorithm capable of real-time performance and requiring no user-given guidance is the challenging problem at hand. The problem of image completion can be formally stated as: given an original image and its corresponding hole mask, automatically fill in the images missing region such that the overall result is coherent and visually plausible with its surrounding information. Figure 1 emphasises this concept. In Figure 1 (a) the area to be removed and re-synthesized is selected

[1] Some authors may have used the term ”inpainting” to mean general image completion. We limit our meaning of completion to mean completion small areas.

[2] During this era, people started to restore medieval artworks bringing medieval pictures up to date by retouching them in attempt to revert deterioration, or to add or remove elements. This is done to preserve the paintings and other fine arts for future generations.



(a) Original Image and unwanted object (red)

(b) Hole mask

(c) Unsatifying Completion Result

(d) Satifying Completion Result

Figure 1.1: Image Completion

(red). A binary mask is then obtained as shown in Figure 1 (b), with white pixels being fully known pixels, black pixels being unwanted/unknown information and gray pixels emphasising the boarder of the image.

In order to recover the image of Figure 1 (a), the operation requires restoring the missing region of the image with known information by intelligently selecting regions from the known part of the image and using this information to restore the unknown region. The selection process affects the quality of the image reconstruction. It will lead to unsatisfying completion results if improperly selected or satisfying completion results if selected properly as seen in Figure 1 (c) and (d), respectively. Figure 1 (c) and (d) were generated using adobe Photoshops content aware fill tool. During the last decade, much progress has been made in dealing with the problem of image completion. These modern approaches can be roughly divided into two groups: **diffusion-based** techniques and **exemplar-based** techniques.

1.2 Purpose and Scope of this Thesis

With a long term goal of creating a fully adaptive image completion algorithm used specifically for real-time 3D video completion, the work described in this thesis focuses on the fundamentals of single image completion in accordance with it corresponding depth map. Specifically, we extend the current start-of-the-art exemplar completion algorithm to incorporate additional 3D depth information. We use this depth information to help constrain the completion process while simultaneously completing the missing region(s) in both depth and colour images. Our completion results show that our proposed method could outperform current selected state-of-the-art completion algorithms.

1.3 Contributions

While previous approaches have produced some remarkable results, most have difficulties in completing images where complex structures exist in the missing region due to unconstrained search spaces, user inputted parameters that need to be tuned between images and invalid matching resulting in error accumulation and propagation. Therefore, we propose an adaptive depth guided image completion method that performs structure and texture synthesis for simultaneous colour and depth completion. Our approach builds upon [9] in a manner similar to [10] by making the following contributions:

1. We use a morphological diffusion based approach to estimate the missing depth information, ensuring the recovery of unconnected boundaries (i.e. Structure Information);

2. Building off the foundations of [9], we constrain the search space to the previously diffused depth information and iteratively optimize the synthesized texture;

3. For each iteration, we determine an optimal patch size for each unknown patch by optimizing a collection of edge oriented histograms;

4. We extend the SSD criteria to incorporate an adaptive variance based weighting;

The algorithms of [9] and [10] are discussed in detail in chapter 3 in order to provide the reader with relevant background information and to understand these contributions.

1.4 Overview of This Thesis

The remainder of this thesis is organized as follows: Chapter 2 reviews existing work related to single image completion; Chapter 3 introduces the fundamentals of exemplar based completion; Chapter 4 illustrates our proposed method in detail; Chapter 5 describes our experiments and the results from both qualitative and quantitative user studies; Chapter 6 draws conclusions and discusses possible future work.

Chapter 2

Related Work

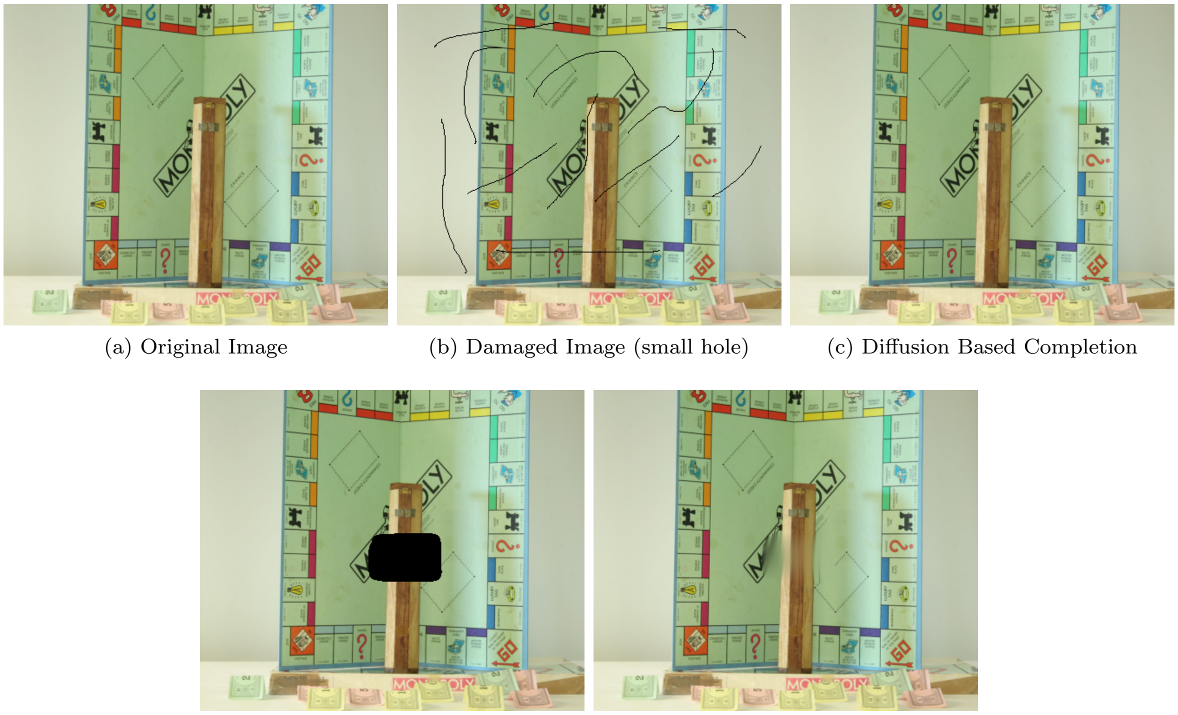
A variety of methods in literature have been proposed for performing image completion. This chapter provides an overview of these methods, along with examples of the kinds of results that they produce. This chapter also outlines the key differences between diffusion-based and exemplar-based image completion.

2.1 Diffusion-Based Methods

Diffusion based inpainting techniques fill in the missing regions through the use of a diffusion process, i.e. by smoothly propagating information along the boundary in-towards the interior of the missing region. This diffusion process is typically achieved by solving a high order, non-linear partial differential equation (PDE). This class of methods was pioneered by Bertalmio et al. [11], who propagated colour information along isophotes according to a third-order PDE. This approach allows for the image intensities to be extended in the direction of level lines arriving at the hole. This results in solving a discrete approximation of the PDE

$$u_t = \nabla^{\perp} u \cdot \nabla \Delta u, \tag{2.1}$$

It was later extended by Ballester et al. who used the PDE in a variational framework [12]. Their variational approach leads to a set of two coupled PDEs, one for the gray levels and the other for the



(a) Original Image (b) Damaged Image (small hole) (c) Diffusion Based Completion

(d) Damaged Imaged (large hole) (e) Diffusion Based Completion

Figure 2.1: Diffusion Based Completion Results

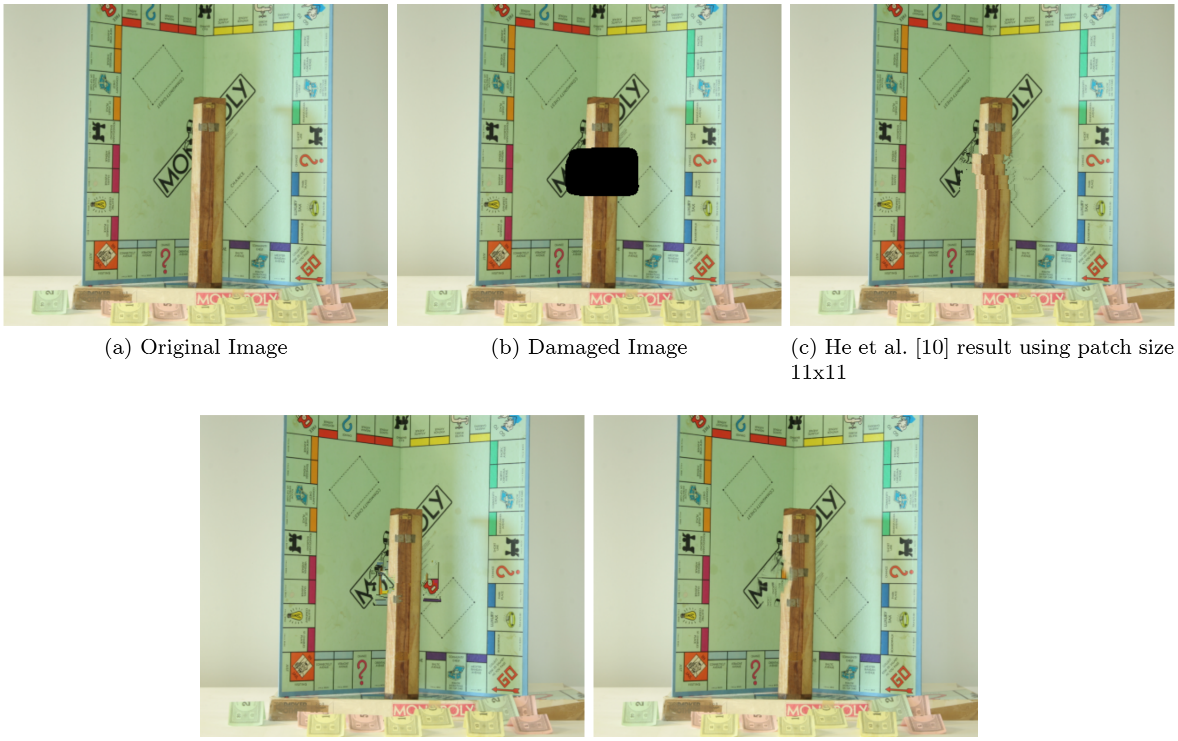
gradient orientations which is solved using its steepest decent. Bertalmio et al. used the Navier-Stokes equations for an incompressible fluid for the completion guidance [13], where the image intensity function plays the role of the stream function whose level lines (isophotes) define the stream lines of the flow. These PDE-based approaches are less effective in handling large missing regions due to its inability to synthesize texture. When the unknown region is quite large, PDE-based methods produce very blurry results. This is shown in Figure 2.1, where the diffusion based completion algorithm of [11] is used for the restoration.

In an attempt to overcome this issue, Berlatmio et al. separate an image into two components [14]: structure and texture . The first component is filled using the PDE based approach given in [11] while the second is filled using a texture synthesis method [14] and the resulting image is a combination of the two. Chan and Shen [15] handle curve structures by incorporating an elastic equation. While Guo et al. [16] perform the diffusion process through the use of a morphological erosion operation. A structuring element is iteratively applied to erode all pixels at the boundary of the hole through structure/texture matching defined within a search space around the hole. They show their results complete some structure features and perform better than PDE based approaches. Similarly, Jawas et al. [17] use morphological dilation to constrain the erosion operation to only consider nearby pixels in the hole filling process.

2.2 Exemplar Based Methods

Exemplar based techniques aim in filling in the missing region by searching for an optimal exemplar pixel/patch located in the source region for each unknown pixel/patch in the target region. These methods are known as texture synthesis methods since they complete desired texture information as shown in Figure 2.2. This class of methods was first introduced by Efros et al. [18], who formulated the problem as a discrete Markov Random Field (MRF) global optimization problem. Spatial coherence is then naturally ensured through a global optimization of MRF energy functions over the entire image. Komodakis et al. [19] optimize the MRF by using belief propagation while Pritch et al. [20] optimize through the use of graph cuts. The shift-map of [20] computes a vector field described in [21], which maps each pixel in the hole to a pixel in the known part of the image. An optimal shift map is then obtained as a graph labelling optimization minimizing a global cost function. He et al. [22] form shifted versions of the input images based upon dominant offsets, and these shifted images are then combined by optimizing a global MRF energy function. The combination of the shifted images are directly used as the patch offsets to fill in the hole. These MRF-based methods can produce visually pleasing and coherent results when the hole to be filled has homogeneous texture and very few structures around it.

The missing regions, in general, may be composed of complex textures and structures, and it has been observed that it is important to separate these two components and start by first recovering the desired structure. Criminsi et al. [23] proposed a patch processing order in which patches on structures are filled in first. The proposed priority order is to emphasize the propagation of structure information, and the filling order is incorporated to prefer the ones with high structure information (i.e. data term) and more available surrounding information (i.e. confidence term). Giving a higher priority to patches which are on the continuation of strong edges leads to a better recovery of unconnected boundaries. Once the patch with a maximum priority is determined, the next step is to perform texture filling by searching the known region for the best matching patch according to a minimum SSD criteria. Huang et al. [24] extends [23] to include a disparity priority in order to improve the matching criteria. Although [23] and [24] performs texture and structure synthesis, matching errors may result and propagate throughout the filling process



(a) Original Image

(b) Damaged Image

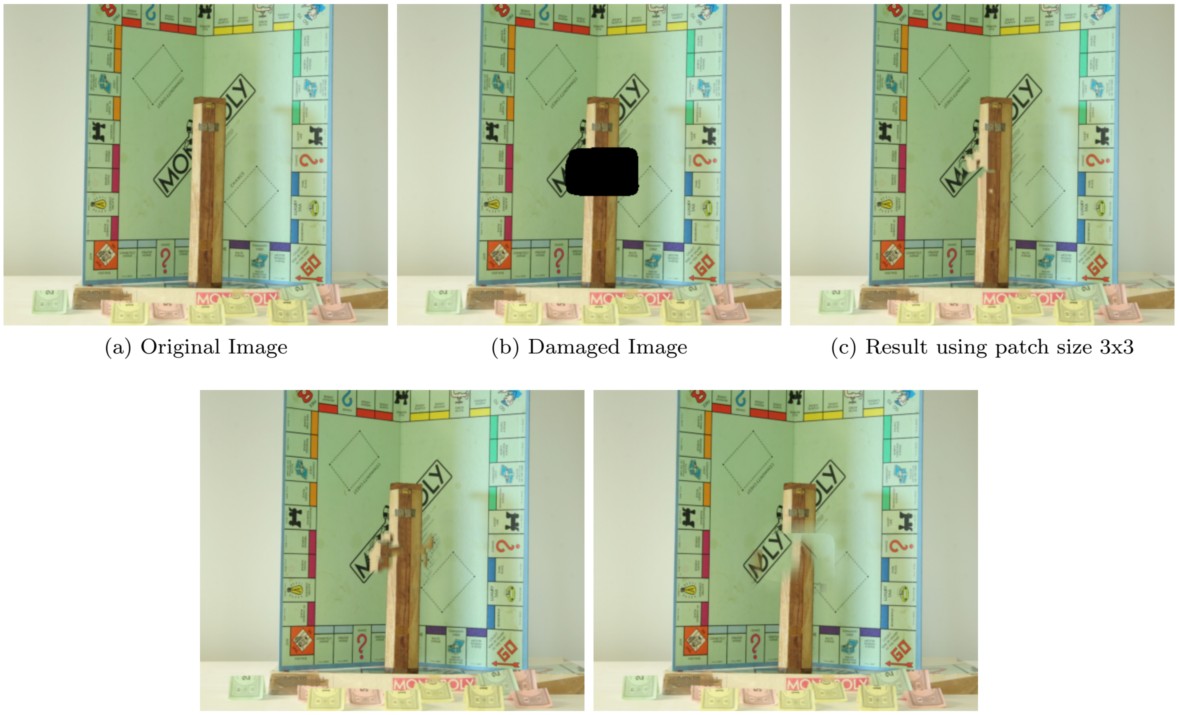
(c) He et al. [10] result using patch size 11x11

(d) Criminisi et al. [23] Result using patch size 11x11

(e) Content aware fill [1] Result using patch size 11x11

Figure 2.2: Exemplar Based Completion Results

since the correlation between patches is not considered. To overcome this issue, Liu et al. [25] proposed an improved SSD criteria, by selecting the best fit candidate using an additional variance term between patches. Komodakis [26] and Wexler et al. [27] proposed a global optimization approach in order to minimize visual inconsistencies and maximize visual coherence. Simakov et al. [28] propose a gradual resizing approach where the completion is performed from course-to-fine. At each resizing step, the global optimization function of [27] is minimized and the colour is updated using an iterative update rule. Barnes et al. [9] introduced a new fast randomized algorithm for finding good correspondences, under the assumption that there is a good chance at least one good match can be found in a randomly assigned approximate nearest neighbor field (ANNF). A combination of [9] and [27] is implemented as the content aware fill in Adobe Photoshop (CS5) [1], which is arguably known as the current state-of-the-art region filling algorithm in terms of visual quality and convergence speed. The completion algorithm implemented by Adobe proceeds in a gradual resizing, similar to [28], however instead of searching every pixel for a match, they use their PatchMatch algorithm to find the correspondences. With additional



(a) Original Image (b) Damaged Image (c) Result using patch size 3x3

(d) Result using patch size 7x7 (e) Result using patch size 25x25

Figure 2.3: Varying Patch Size Completed using [1]

depth information made available, He et al. [10] further extended [9] to simultaneously in-paint both colour and depth images for object removal. They constrain the search region to regions farther in depth than the object being removed. This method performs well for simultaneous colour and depth texture synthesis but fails in the recovery of linking similar distinct areas across the hole (i.e. structure information). Hung et al. [29] use a Laplacian pyramid to simultaneously inpaint structural features and texture features for each level in the pyramid.

In all previously mentioned methods, the patch size is fixed and most often specified by the user [9,10, 23,27]. Criminisi et al. [23] suggests that the patch size be slightly larger than the largest distinguishable texture element present within the known region. However, it is very difficult to choose a good patch size without trial and error and it may be inappropriate to let all the patches have the same size because local feature characteristics change within the image. Figure 2.3 shows results completed using [1] and using different patch sizes. As can be seen in Figure 2.3 the patch size directly affects the overall completion accuracy. To address this issue, Drori et al. [30] proposed a fragment-based method which proceeds in a multi-scale fashion approximating the unknown region with adaptive image fragments (patches).

The patch sizes are chosen to be inversely proportional to the spatial frequency. This method performs well in many cases, but sometimes introduces blurring artifacts. Zhou et al. [31] introduce an adaptive patch completion algorithm, based on local image gradient magnitude histograms. They formulate an optimization problem consisting of gradient magnitude histograms in accordance with a saliency map computed in [32] to identify which elements in a scene are likely to attract the attention of human viewers. The optimal patch size for all source patches is then the solution to a minimization problem. The filling of source pixels is then achieved using the method of [23]. This method requires the use of a visual saliency map to emphasize distinct and homogeneous features located within the image and is used in accordance with each local gradient magnitude histogram calculation. This method performs well in limited cases, when the object being removed is salient (i.e. stands out from its neighbors) but may produce unsatisfying completion results when the object being considered is non-salient.

Chapter 3

Exemplar-Based Completion

The core of our proposed method builds directly on the single image PatchMatch-based completion of Barnes et al. [9] and in a fashion similar to He et al. [10]. This chapter describes the foundations of exemplar-based completion as well as the efficient PatchMatch searching algorithm used to optimize the search. In the last section we explain how the PatchMatch algorithm is used in accordance with the exemplar selection to perform image completion. This algorithm is the backbone to our improved method.

3.1 Exemplar Selection

Given a colour image I and its corresponding depth map D, which can be obtained by active sensing, calculated using a stereo correspondence algorithm [33] or generated using user provided labelling [34,35]. Let $H_I \subseteq I$ be a user-specified hole region within the original colour image and let $H_D \subseteq D$ be the same hole region located within the depth map. Therefore, H_I and H_D denote all unknown points(the target region) within I and D respectively, and $H_I = H_D = \Omega$ denotes all unknown pixels. Furthermore, let $\Psi = H_I^c = H_D^c$ be the known remaining pixels (the source region) located outside the hole and separated by some boundary $\partial\Omega$ between the known and unknown regions.

Given both Ψ and Ω, our goal is to complete the missing region Ω with some new information Ω^*, ensuring that the resulting image will have as much global visual coherence with its surroundings Ψ. We say that, Ω has global visual coherence with Ψ if every local patch in Ω can be found somewhere in

Ψ and if every patch found is the best matching patch. Therefore, we seek new information Ω^* which minimizes the following objective function:

$$Coherence(\Psi, \Omega^*) = \sum_{p \subset \Omega} \underset{q \subset \Psi}{min}\, d(\psi_p, \psi_q), \tag{3.1}$$

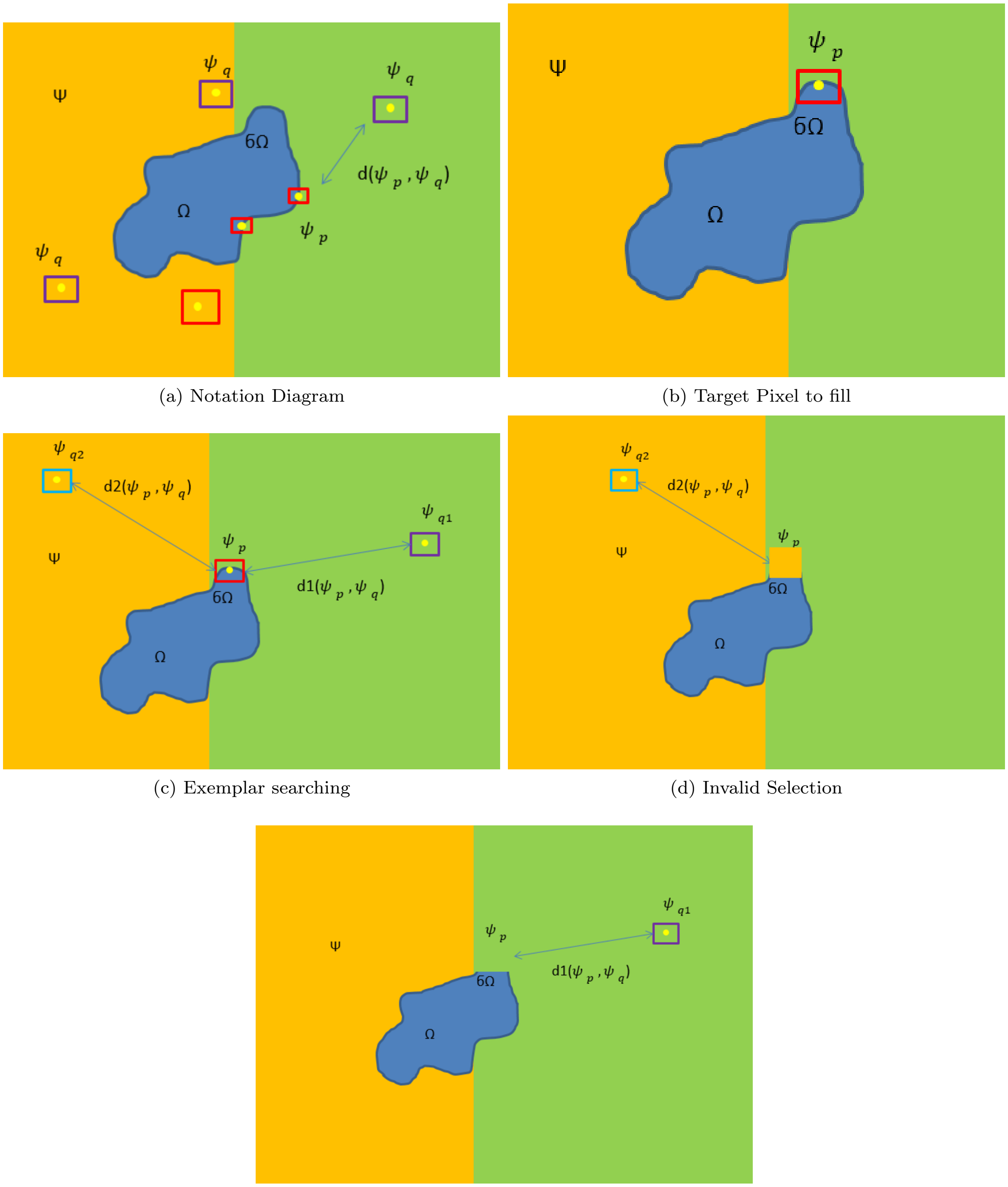
where Ω is the target/missing region, Ψ is the source/known region, ψ_p is a square patch centered at pixel p with coordinates (x, y) and with $p \in \Omega$, ψ_q is an exemplar selection patch centered at pixel q with coordinates (x, y) and with $q \in \Psi$ and $d(\psi_p, \psi_q)$ is a measure of the difference between patches ψ_p and ψ_q.

To ensure maximum visual coherence in equation (3.1) we need $Coherence(\Psi, \Omega^*)$ to be a minimum. Therefore, we need to ensure that every patch located with the missing region is filled with its best matching patch located within the known region, thus penalizing any unwanted artifacts. The optimal exemplar we are searching for must satisfy

$$\psi_q = arg \underset{\psi_q \subset \Psi}{min}\, d(\psi_p, \psi_q), \tag{3.2}$$

for each $p \in \Omega$. Figure 3.1 (a) shows a Notation Diagram for exemplar-based completion. Figure 3.1 (b),(c),(d) are used to explain the selection process. Starting with Figure 3.1 (b), we first select a target pixel and its corresponding patch ψ_p with $p \in \Omega$, our goal is to fill in the unknown information inside the red patch. In the red patch, there contains known and unknown information and the comparison (i.e. $d(\psi_p, \psi_q)$) is only done between known pixels. Figure 3.1 (c) shows two selection patches labelled ψ_{q1} and ψ_{q2}. If we select ψ_{q2} as our match then the yellow information will be propagated into the unknown region and the result will be non-coherent with respect to the source region as shown in Figure 3.1 (d). This is because when performing a distance measurement between patches ($d(\psi_p, \psi_{q2})$) the result is very large indicating non-similarity. [1] If we select ψ_{q1} as our match the green information will be propagated into the unknown region resulting in a coherent result as shown in Figure 3.1 (e). This is due to very similar matching between ψ_p and ψ_{q1}. For this reason, if equation (3.2) is satisfied for every $p \in \Omega$ then we ensure maximum visual coherence with the surrounding information. It is also important to note that some algorithms do not simply perform direct copying and pasting of pixels into the unknown region but perform colour updating based upon a weighting function [1, 27].

[1] In most algorithms [1, 23] this distance measurement is the sum of squared difference in the RGB colour space.



(a) Notation Diagram

(b) Target Pixel to fill

(c) Exemplar searching

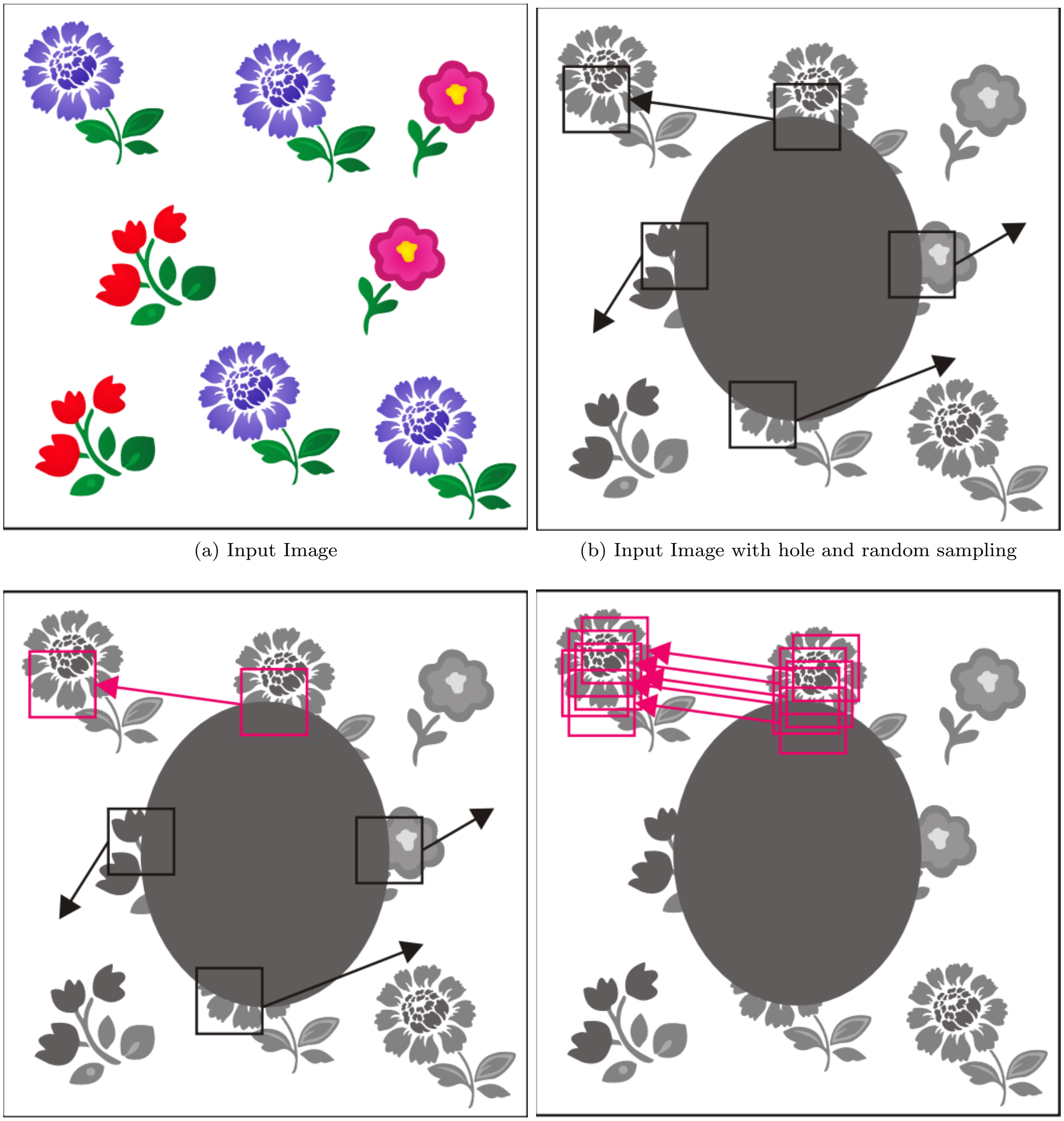
(d) Invalid Selection

(e) Valid Selection

Figure 3.1: Exemplar-Based Completion

3.2 PatchMatch Searching

Matching patches between two images (or in our case the source and target region), is also known as computing the nearest-neighbour field which has been proven as a useful technique in the exemplar selection. This nearest-neighbour field contains the offsets to the matching patches. PatchMatch, introduced in [9], represents an approach to search the image space in attempt to find optimal exemplars for each unknown pixel, which implies an optimal nearest neighbour field. In contrary to exhaustive searching, PatchMatch is based on random sampling and propagation of good guesses. The main idea behind PatchMatch is the approximate nearest neighbour (ANNF) algorithm. Even though it does not guarantee to find an exact nearest neighbour, it converges very fast and the found ANNF corresponds to the nearest neighbour field found by exhaustive searching after several iterations [9]. In real-world digital images, single pixels are rarely a feature on their own and are usually apart of a larger coherent region. Therefore, neighbouring pixels will most likely have their nearest neighbours placed alongside. Taking this into consideration, the key insights driving the algorithm are that good patches can be found via random sampling and due to the natural coherence of images, these matches can propagate quickly to the surrounding area. Figure 3.2 demonstrates this idea. In Figure 3.2 (b) we have the hole region marked in gray. Further, in Figure 3.2 (b) we perform random sampling in attempt to find good matches for the four target patches. While these random guesses are likely to be incorrect most of the time, Barnes et al. [9] state that in a sufficiently large region, a few lucky guess will be almost the correct matching. Figure 3.2 (c) shows a good guess for a patch is found. It is likely that many nearby patches have similar matching patches as shown in Figure 3.2 (d) and offsets are propagated.



(a) Input Image

(b) Input Image with hole and random sampling

(c) Good Guess is found from random sampling

(d) Nearby patches are similar correspondances

Figure 3.2: PatchMatch Algorithm

3.2.1 Computing the Approximate Nearest Neighbour Field

The algorithm has three main components in determining the ANNF.

Initialization

The nearest-neighbour field can be initialized by randomly assigning random values to the field or by using prior information. In this step we randomly guess where good matches may be found.

Propagation

If we denote the coordinate of p as (x, y) and the offset between p and its exemplar q as $v(x, y)$ we attempt to reduce $D(v(x, y))$, where D is a distance function between two patches, by exploiting the spatial coherence with its surrounding neighbours. On odd iterations, we attempt to improve $v(x, y)$ using the known offsets of $v(x - 1, y)$ and $v(x, y - 1)$ and taking the new value of $v(x, y)$ to be

$$\underset{(x,y)}{\operatorname{argmin}}(D(v(x,y)), D(v(x-1,y)), D(v(x,y-1))). \tag{3.3}$$

On even iteration we search in the alternate directions and attempt to improve $v(x, y)$ using the known offsets of $v(x + 1, y)$ and $v(x, y + 1)$ taking the new value of $v(x, y)$ to be

$$\underset{(x,y)}{\operatorname{argmin}}(D(v(x,y)), D(v(x+1,y)), D(v(x,y+1))). \tag{3.4}$$

This means that we compare each patch in the offset map with its left and upper patches on even iterations and right and bottom patches on odd iteration. The patch which is more similar to the according target patch is the one which replaces the offset values at the belonging position in the offset map.

Random Searching

After the propagation step we preform random searching in which we search for good patches from offsets around a certain region around the current pixel. If there is a better match than the offset map gets overwritten at this position. Therefore, we improve $v(x, y)$ by testing a sequence of candidate offsets at an exponentially decreasing distance from $v(x, y)$. Letting $V_0 = v(x, y)$, we search inside the radius dedicated by the formula:

$$U_i = V_0 + W\alpha^i R_i \tag{3.5}$$

Where W is the search window, i is the iteration, α exponential factor which decrease the size of the search area every iteration, R_i random value between [1,-1] and V_0 is the center of the radius. A a value of $\alpha = 0.5$ is proposed by [9].

Halting Criteria

As shown in [9], a fixed number of iterations had been found to work well. Barnes et al. show that after approximately 5 iterations that NNF had almost always converged.

3.3 Adobe Content Aware Fill

As briefly mentioned in Chapter 2, the image completion algorithm of Adobe photo-shop is considered the current state-of-the-art method and its algorithm proceeds as follows.

1. Using a coarse-to-fine gradual resizing process to further enforce global consistency and to speed up the convergence, they perform an iterative process in multiple scales using spatio-pyramids[2].Each pyramid level contains half the resolution in spatial dimensions and the optimization starts at the coarsest pyramid level with the solution being propagated to finer levels for further refinement. At the coarsest level, all missing pixels now contain diffused colour information. During the resizing process, the original image and hole mask are down sampled to the current resolution.

2. It uses PatchMatch to search for matches between the missing pixels and the known region:

2a. PatchMatch initializes its pixels randomly, that is the ANNF contains a random sample of offsets from the source region of the original image.

2b. It updates the nearest neighbour field using propagation and random search. In the comparisons between patches, it uses the sum of squared differences in the RGB colour space to determine optimal exemplars and to ensure equation (3.1) is minimized.

2c. Steps 2a and 2b may be iterated multiple times for better results. Barnes et al. [9] show it converges in 4-5 iterations.

[2] The authors do not provide detail as to which downscaling method was used.

3. Once the NNF converges, the algorithm then updates the colour information, using a voting algorithm. Let c be the colour value of a target pixel $p \in \Omega$. After running several iterations of PatchMatch which ensures proper convergence, the colour value of pixel p is then updated using a weighted blending of values of the sources patches ψ_q matched to each target patch ψ_p that overlaps pixel p. The colour of pixel p is then given by

$$c = \frac{\sum w^i c^i}{\sum w^i}, \tag{3.6}$$

where c^i is the colour value of pixel p given by a source patch ψ_q^i and w^i is a per-pixel weighting. This same distance transform weighting function as in [27], defined as

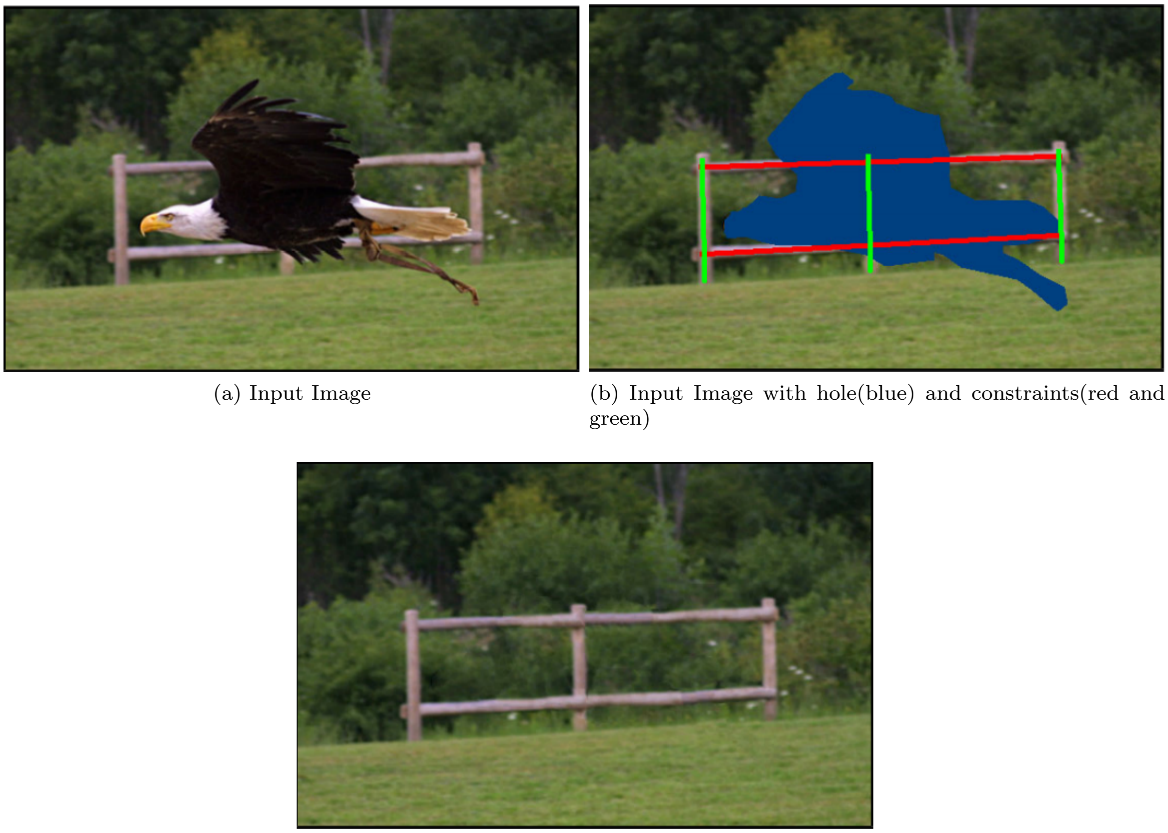
$$w^i = \gamma^{-dist(p,T)}, \tag{3.7}$$

As proposed by Wexler et al. [27] we use $\gamma = 1.3$ and $dist(p, T)$ is the spatial distance from p to the boundary of the target region T.

4. When the results are converged at a given resolution, they are then up-scaled and we start over again at step 2. This repeats until the image is up-scaled to its original size.

Figure 3.3[3] shows a result produced using this method. Due to the randomization of offsets in each level of the pyramid, this algorithm suffers from unconstrained search spaces. In order to ensure proper structure propagation in the completed image, user provided constraints must be manually drawn on the image as shown in red and green of Figure 3.3 (b). Without these constraints, the fence in Figure 3.3 (c) would not be completed properly. Further, in the matching of patches, the size of the patch is also a user inputted parameter, the user must manually input a desired patch size and using trail and error visually determine which patch size produces the best completion result.

[3] Result taken from [9]



(a) Input Image

(b) Input Image with hole(blue) and constraints(red and green)

(c) Completed image using [1]

Figure 3.3: Content Aware Fill Result

Chapter 4

Methodology

We first allow to user to select an unwanted region (or multiple regions) located within a colour image. In the filling process, we modify the conventional exemplar based completion methods by extending to a RGBD framework similar to [10]. We first fill in the depth map in the unknown region and constrain the PatchMatch-based search space to only consider pre-filled depth information in the texture synthesis of the colour image. In this chapter we provide further detail of the depth guided image completion approach, by separating the completion process into depth completion and colour texture synthesis.

4.1 Depth Completion

A depth map is stored as grey scale image, where the intensity values represent the depth, or distance, of a pixel from a camera. When working with depth information, it is important to note that a depth map has very different characteristics from an ordinary RGB image. Unlike colour images, which contain rich texture information, a depth map generally contains smooth regions with strong edges that coincide with distinct object boundaries. Unless the user chooses to retain the existing depth that information, we then first fill in the depth maps in the respective target region using a diffusion filling algorithm that ensures any structure information is properly propagated across the target region.

4.1.1 Pre-processing of the depth map

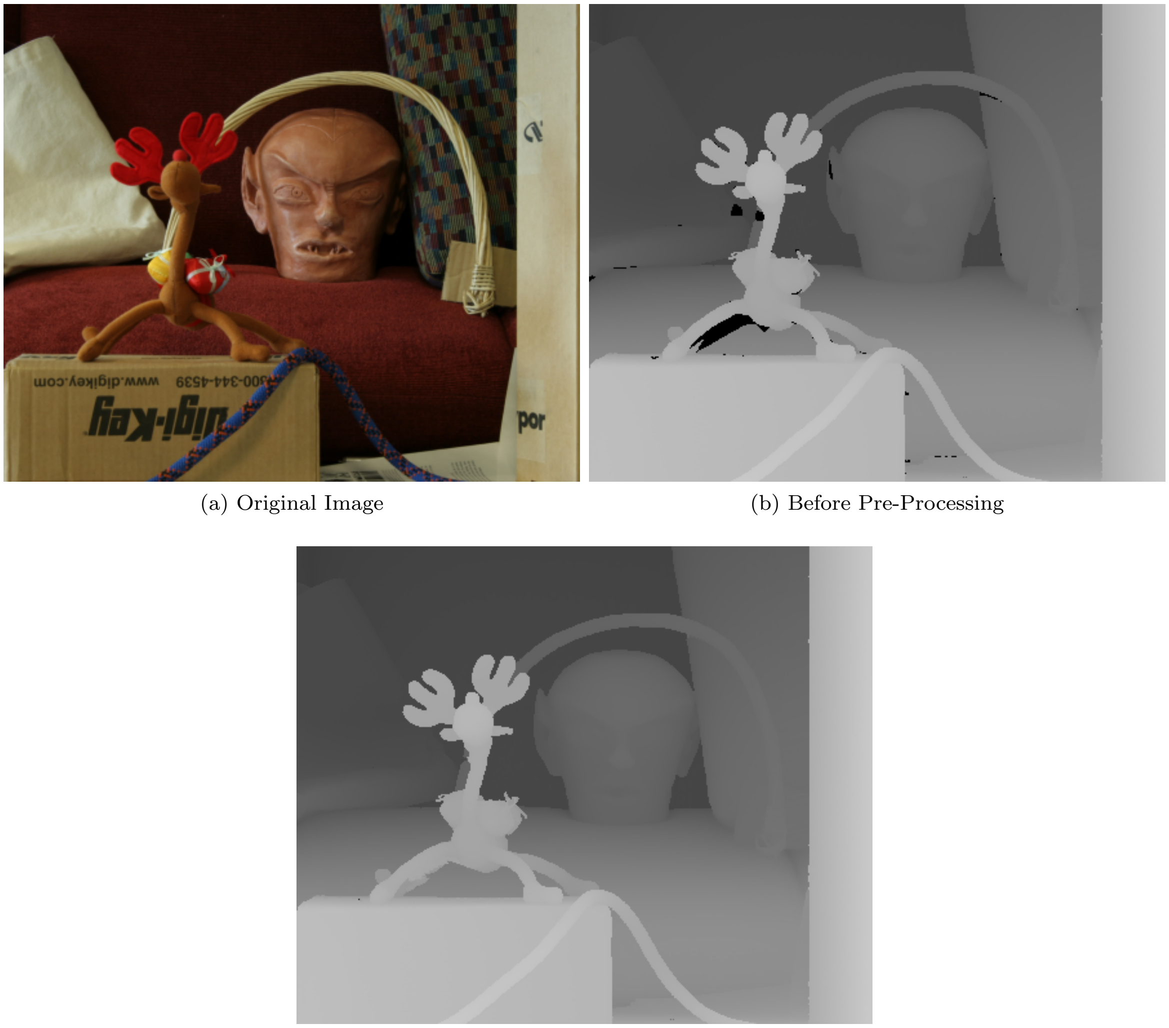
Some form of pre-processing is necessary when working with depth maps as they often have missing pixels due to occlusions or spurious depth estimates due to limitations of the sensor acquisition tools [33]. Therefore, prior to use, we first fill any missing regions in the depth map using the hole filling method of [36] and [10]. For each missing depth pixel we consider its 8-connected neighbours as candidates for the filling process. Since missing depth information is often a result of when a foreground object partly occludes a background object, we assume that the missing depth information is to be propagated from its adjacent background [10].We define the foreground object as the pixel with the largest intensity value and the background object as the pixel with the smallest intensity value contained in the 8-connected neighbourhood. We then replace the unknown pixel with the smallest value, and repeat for all unknown points. That is, for every $p \in \Omega$ (Where Ω in this case are missing pixels due to occlusions), there are eight candidate depth values considered for filling, which is denoted as $N_8(p)$. Using our assumption, we select $min(N_8(p))$ as the depth value for pixel p. Figure 4.1 (b) and (c) shows the before and after pre-processing of the depth map.

4.1.2 Depth Map Completion Via Diffusion

After the depth map pre-processing step, we use a method similar to [17] and [16]. For each patch ψ_p centered on $p \in \Omega$, we iteratively erode and restore all the pixels on the current boundary of Ω through a feature matching algorithm propagating known pixels smoothly into the unknown region. The algorithm proceeds as follows: First, within a search scope $\Psi_D = \Omega \oplus B$, where $\oplus$ denotes the morphological dilation operation, and B is a 7x7 structuring element, each known patch ψ_q with $q \in \Psi_D$ is compared to ψ_p with $p \in \Omega$ using the distance measurement.

$$d(\psi_p, \psi_q) = d_{depth} = \sum_{(i,j) \in N_8(p,q)} \left| \psi_p(i,j) - \psi_q(x+i, y+j) \right|, \tag{4.1}$$

where ψ_p is a square patch with grey level intensities $\psi_p(i,j)$ at location (i,j), ψ_q is an exemplar square patch with grey level intensities $\psi_q(i,j)$ at location (i,j), $N_8(p,q)$ is an 8-connected neighbourhood centered at p and q and d_{depth} is the sum of absolute differences between ψ_p and ψ_q in depth. It is important to note that only valid pixels contained in Ψ are used in the computation of d_{depth} and we



(a) Original Image

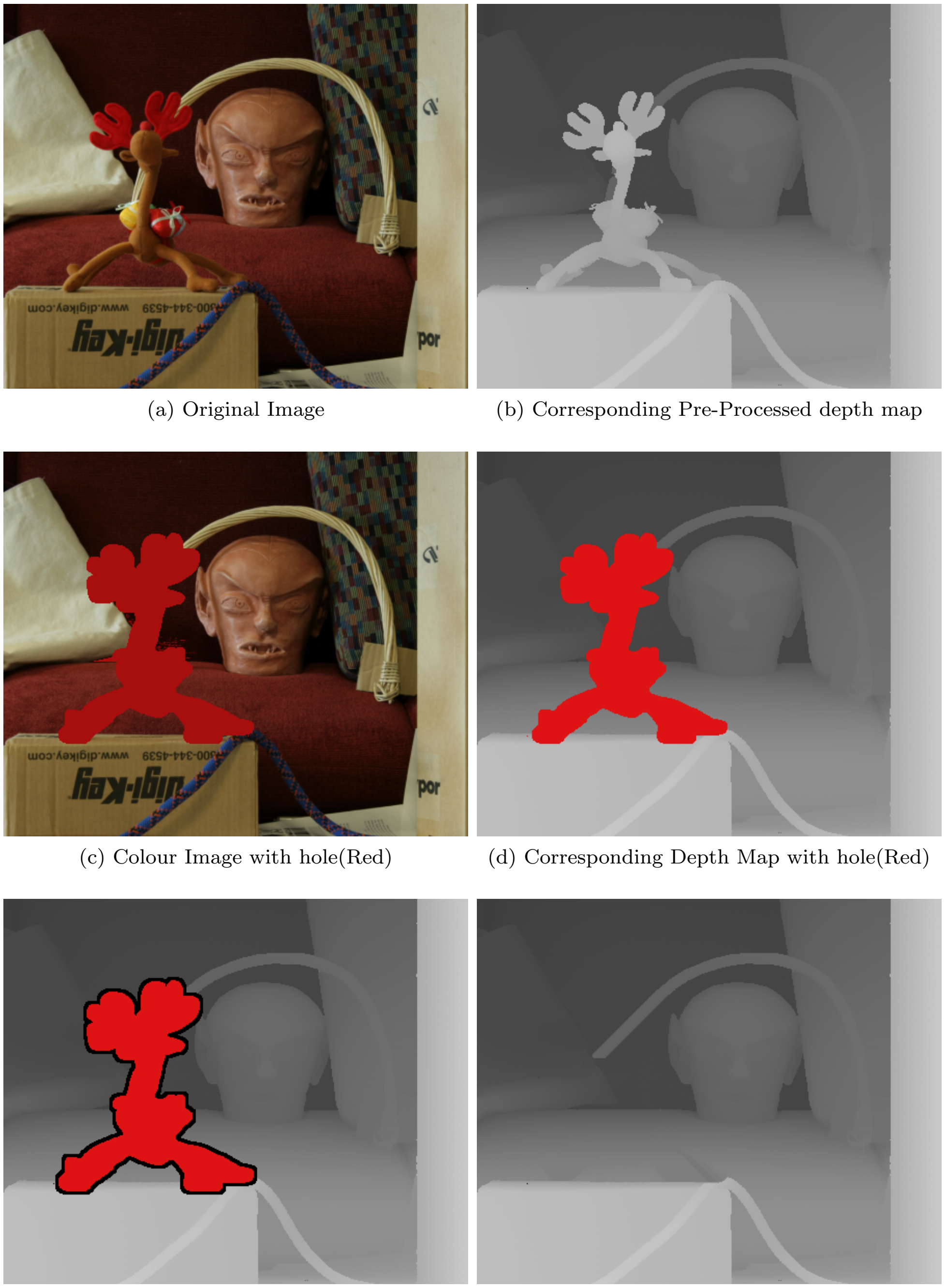
(b) Before Pre-Processing

(c) After Pre-Processing

Figure 4.1: Depth Map Pre-Processing

normalize d_{depth} by dividing by number of valid pixels. Once the patch ψ_q is found within the search scope and satisfies equation (2.3), the values of the known pixels within ψ_q are then copied into the unknown patch ψ_p and this process repeats for all $p \in \Omega$. After each iteration, we repeat until each unknown pixel is flagged as known. Assuming distinct image areas are spatially continuous and are only separated by the hole, this algorithm provides diffusion from nearby pixels around the hole to be diffused into the hole propagating both structure and texture.

Figure 4.2 (a) and (b) show the original image and its corresponding pre-processed depth map, (c) and (d) show a user selected region marked in red (i.e. Removal of a foreground object) for the colour image and depth map, respectively. For each unknown patch ψ_p centered with $p \in \Omega$ (the red region in this case), we constrain the search of candidates exemplars to only look within black region shown in Figure 4.2 (e). This not only speeds up the completion process (i.e. As opposed to searching trough the entire image) but ensures structure is properly propagated across the target region as shown in Figure 4.2 (f).



(a) Original Image

(b) Corresponding Pre-Processed depth map

(c) Colour Image with hole(Red)

(d) Corresponding Depth Map with hole(Red)

(e) Constrained Search Space (black)

(f) Diffused Depth Information

Figure 4.2: Diffused Depth Information

4.2 Colour Image Texture Synthesis

After the depth completion, we use the PatchMatch-based algorithm described in chapter 3 to find optimal exemplars within the source region. We use the filled depth information as a constraint narrowing down the search field. Different from [9] we do not proceed in a multi-scale fashion but preform the PatchMatch steps directly on the image containing the missing information.

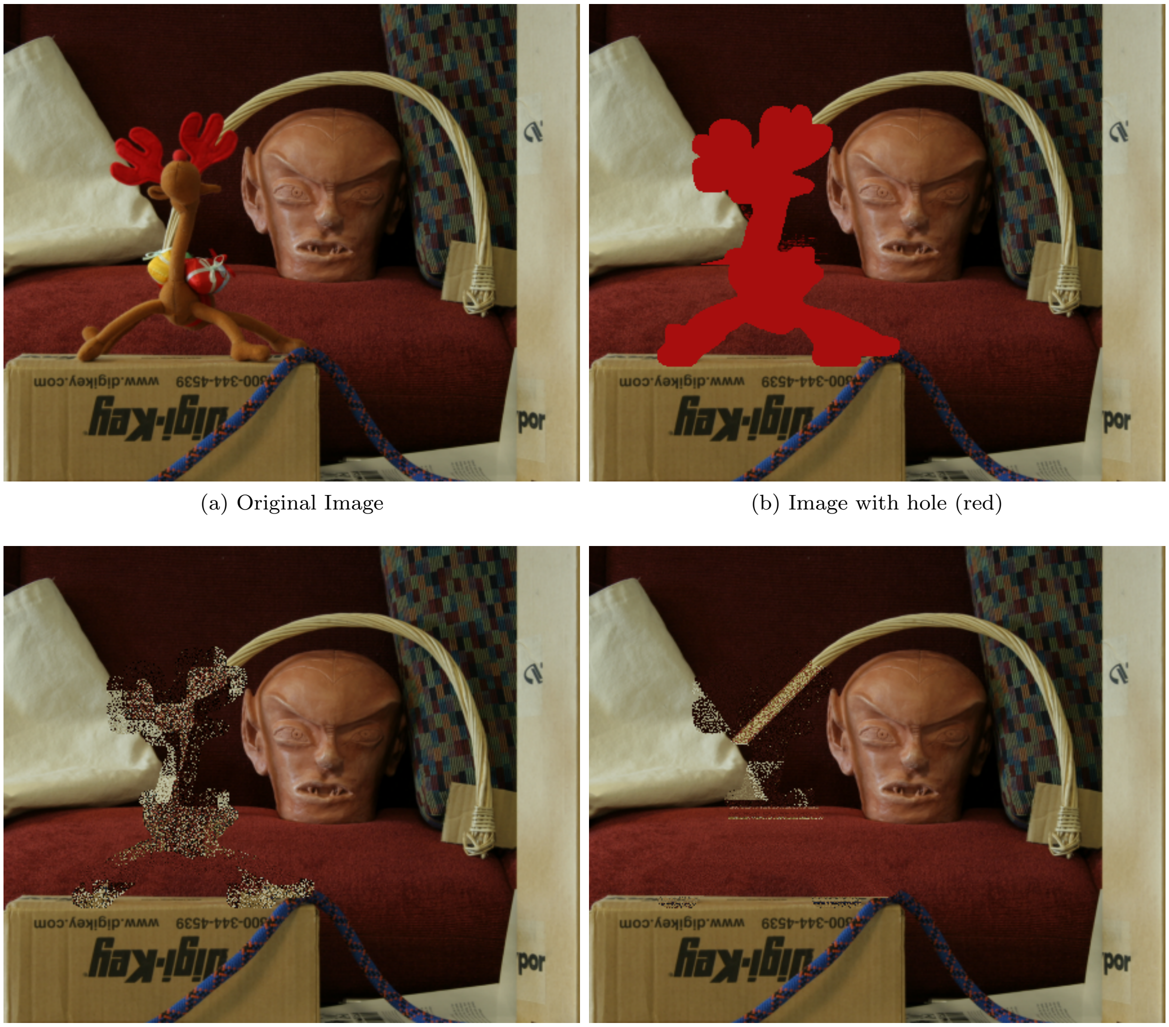
4.2.1 Initialization

Before we start the ANNF for the first time, it is necessary to initialize the offset map with random values. We differ from [9] in that we do not allow the initialization to be completely random and we differ from [10] since we do not initialize to only farther depth values. We use the completed depth information to constrain the initialization of the ANNF allowing for a better initialization.

After completing the desired depth map, each unknown pixel in depth now contains a new depth value satisfying equation (4.1). As opposed to [9] and [10] we use this completed depth information to constrain and initialize our ANNF for all pixels within Ω by randomly selecting colour information corresponding to its filled depth information. In other words, for each unknown pixel $p \in \Omega$ we randomly select

$$p = \{q_R | q_R \in \Psi \cap q_R \in d_p\}. \tag{4.2}$$

where q_R is a randomly selected exemplar, and d_p is the completed depth information for pixel p If two objects have the same depth information but have very different colour characteristics (i.e. colour or texture), we can further constrain the search space and avoid invalid random selection by adding an additional closeness constraint. We allow the offsets to only be selected in an area around the hole. This allows only searching within the region under consideration, and avoids searching within other regions at the same depth information. Figure 4.3 (c) and (d) shows He et al. [10] initialization in comparison to our depth constrained initialization, respectively. Using the depth map from Figure 4.2 (f) as our depth constraint values, it can be seen in Figure 4.3 (c) that our randomly initialized colour values and offsets (i.e. ANNF) enhance the correctness of the randomly selected pixels. The more accurate the Initialization the better the prorogation of offsets in the subsequent steps, allowing for a more coherent and visually plausible completion result.



(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Initialization

(d) Depth Constrained Initialization

Figure 4.3: Colour Image Initialization

4.2.2 Optimization

In the process of selecting the best matching patch, many algorithms take the sum of squared differences (SSD) in the RGB colour space between the target patch(ψ_p) and the exemplar selection patch (ψ_q) as their matching criteria [1, 23]. He et al. [10] extend to consider the SSD in both colour and depth. They choose the patch that results in a minimum SSD value and satisfying equation (2.3) [1, 10, 23]. However, within each patch ψ_p with $p \in \Omega$ there are reliable/known pixels that fall within Ψ and unreliable/unknown pixels that fall within Ω as shown in Figure 4.4 where the red pixels correspond to Ω and everything outside Ω corresponds to Ψ.

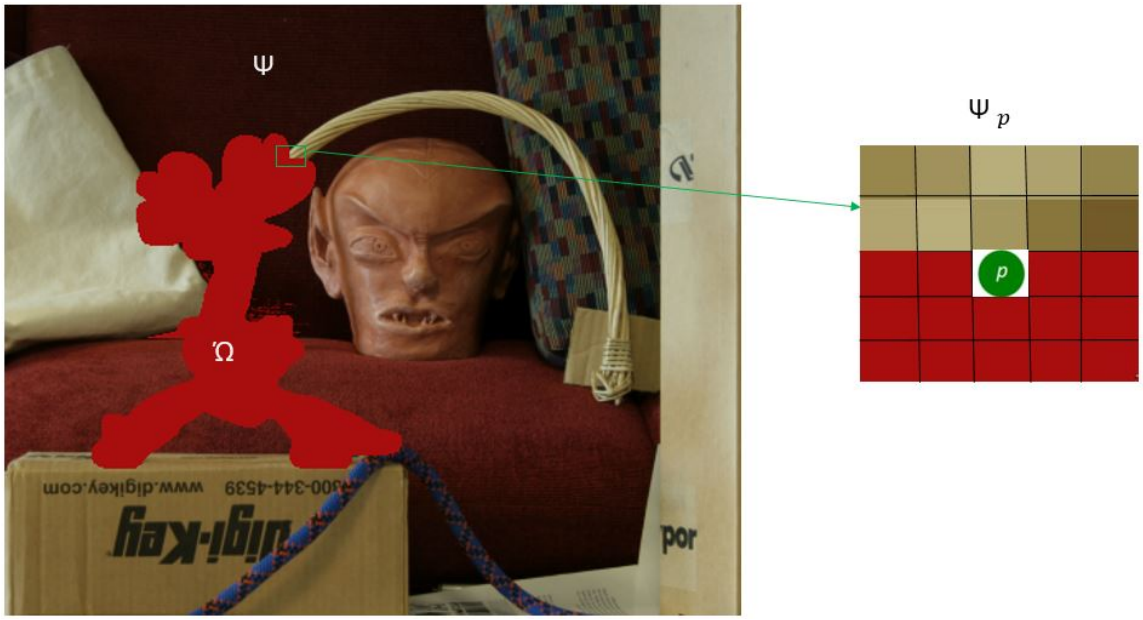


Figure 4.4: Target Patch ψ_p

In previous approaches, this SSD criterion only considers the similarity between the known pixels in the target patch (i.e. pixels that fall within Ψ) and the corresponding pixels in the exemplar selection patch thus, a minimum SSD value corresponds to only a good match of known pixel values [1, 10, 23] as shown in Figure 4.5 where the known pixels in ψ_p and corresponding pixels in ψ_q results in an exact match.

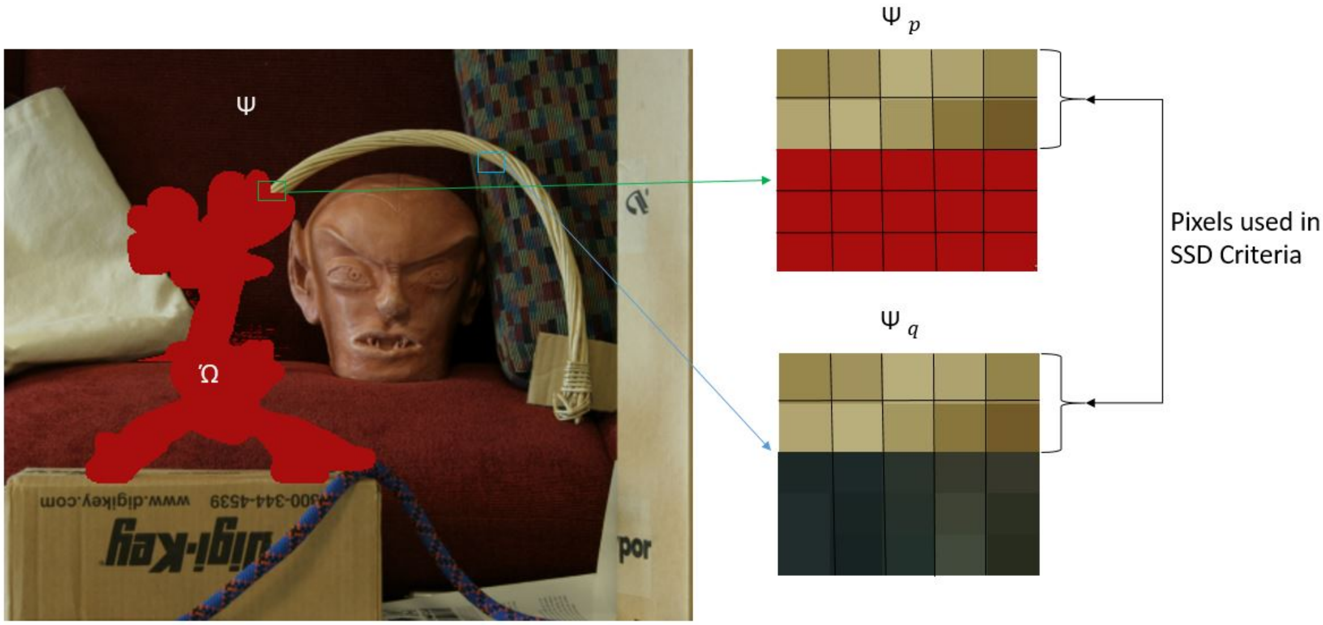


Figure 4.5: Invalid SSD Comparision

Directly replacing (or even using) the blue pixels of patch ψ_q to fill the red pixels in patch ψ_p in Figure 4.5 , results in false local minima and error accumulation and propagation throughout the unknown region since structure not properly completed. This is due to insufficient information being used in the matching criteria and it does not consider the variance between the known target pixels and the pixels that are candidates for replacing the unknown target pixels. Therefore, to overcome this issue, we further improve the SSD objective function of [1, 10, 23] by incorporating a variance dependant weighting, defined as:

$$d(\psi_p, \psi_q) = (1 - T(\sigma^2))d_{colour} + T(\sigma^2)d_{depth}, \tag{4.3}$$

where d_{colour} and d_{depth} are the sum of absolute differences between ψ_p and ψ_q in the RGB colour space and depth respectively, σ^2 is the variance of the known target pixels and the candidate pixels and $T(\sigma^2)$ is a membership function used to penalize colour mismatch. Variance is a measure that determines how far points are spread out relative to their mean value. A small variance indicates that the data points are very close to their mean, while a high variance indicates that the data points are very spread out from the mean. While the randomly selected offsets provide a good idea of where a potential match could be found, its colour may not be coherent with its surroundings which may produce false SSD colour

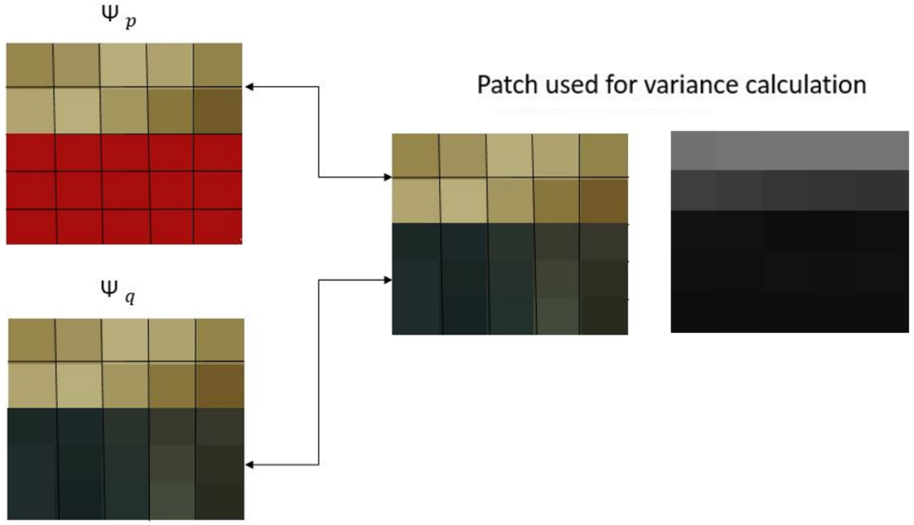


Figure 4.6: Variance Patch

matching and error accumulation. To avoid this, we take the known pixels in the target patch ψ_p and the candidates that will be replacing the unknown target pixels ψ_q and determine the local variance. Since determining the local variance of a RGB image does not result in a single value, we average the three colour channel and determine the local variance in grey scale this is shown in Figure 4.6. If the variance is small we have a good indication that the candidate pixels are a good match and equation (4.3) will prefer d_{colour} over d_{depth}. However, if the variance is high, equation (4.3) then prefers d_{depth} over d_{colour}. We use a membership function $T(\sigma^2)$ which is similar to the soft threshold function of [37] as our weighting which controls the error term in equation (4.4). We define our membership function as:

$$T(\sigma^2) = \frac{1 + tanh(\phi(\sigma^2 - \epsilon))}{2}, \tag{4.4}$$

where ϵ is the threshold value and ϕ controls the sharpness of the transition. Figure 4.7 shows our membership function for different values of ϕ and ϵ. In our experiments we use $\phi = 7$ and $\epsilon = 0.5$ which corresponds to the yellow curve in Figure 4.7. The yellow curve was chosen based upon experimental results, it was concluded that a patch with a variance value of $\sigma^2 = 0.75$ contained a non-coherent match when used to repair the unknown region. Therefore, if $\sigma^2 \geq 0.75$, we use the yellow curve to prefer d_{depth}

over d_{colour} (i.e. $T(\sigma^2) = 1$). It was further concluded that a patch with a variance value $\sigma^2 \leq 0.2$ contained a coherent match when used to repair the unknown region. Therefore, we use the yellow curve to prefer d_{colour} over d_{depth} (i.e. $T(\sigma^2) = 0$).

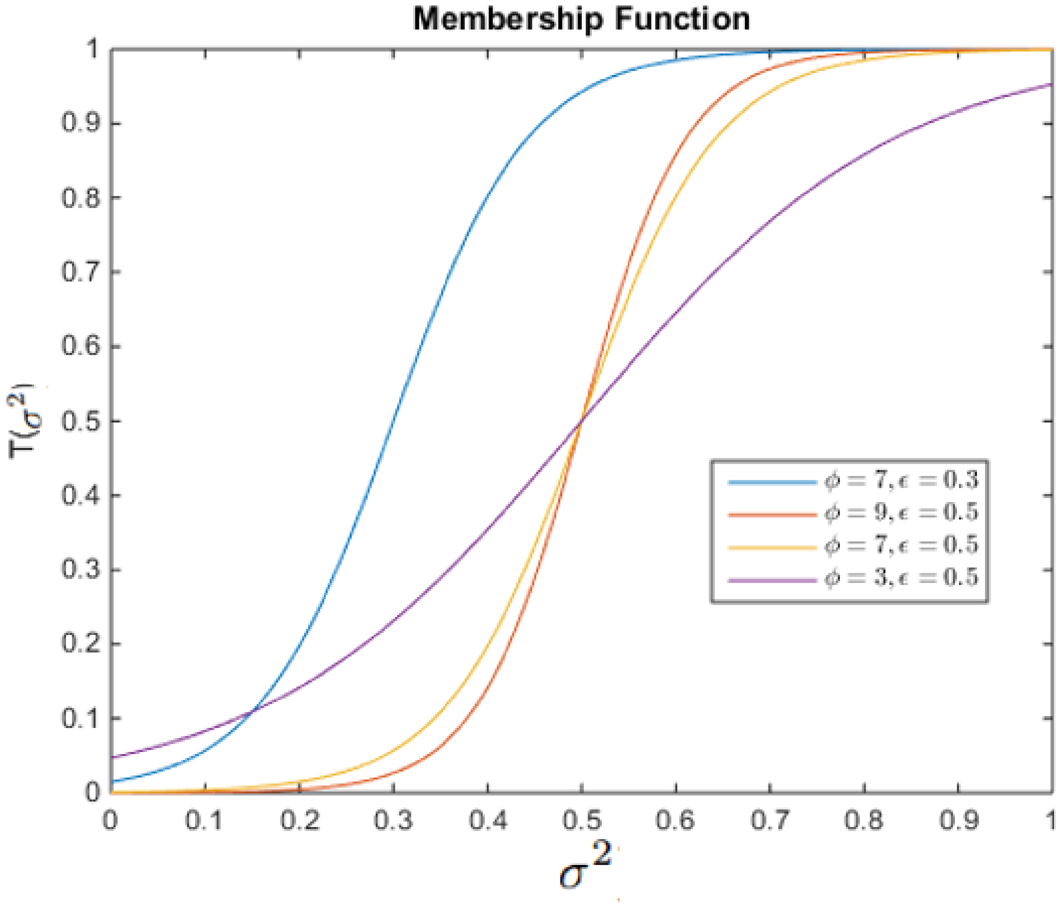
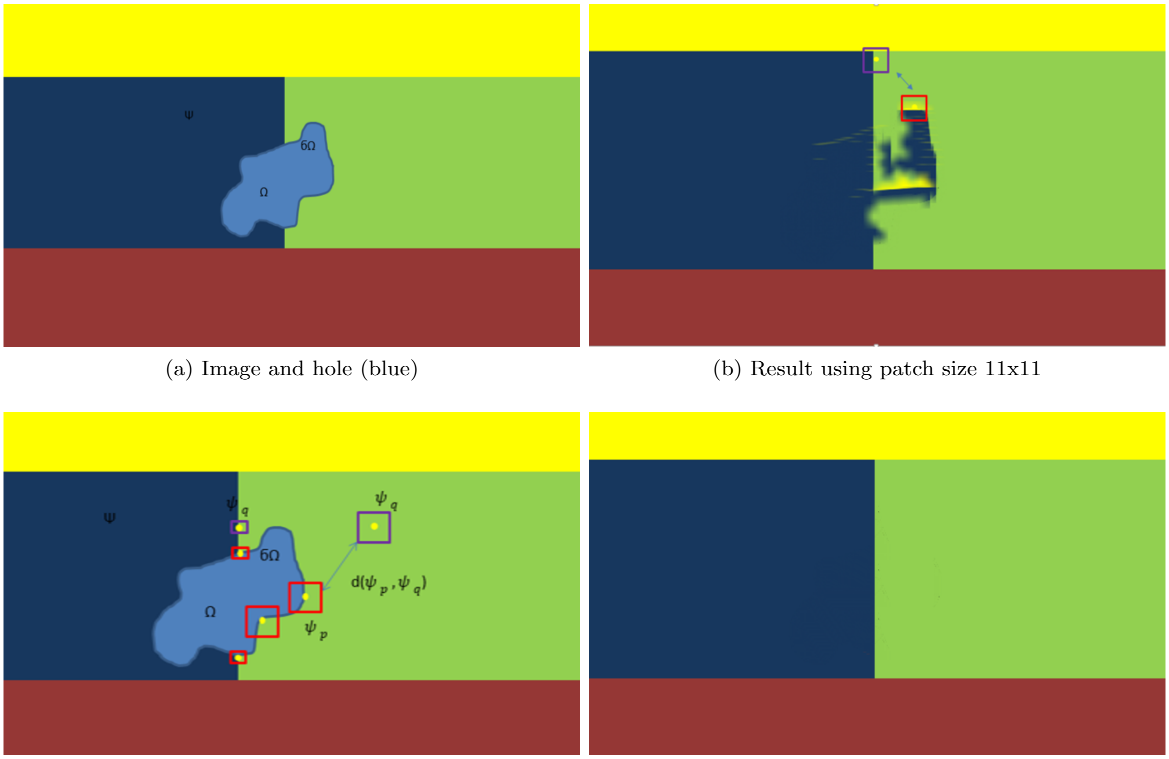


Figure 4.7: Variance Membership Function

For further emphasis of this concept and using the patches from Figure 4.5, we formulate a new patch for the variance calculation as shown in Figure 4.6. Although the SSD calculation may result in a mimima it is clear that the variance of this new patch is very large, implying a false local minima was found. Therefore, we penalize the SSD in colour (i.e. d_{colour}) and avoid error accumulation and propagation into the target region.

4.2.3 Adaptive Patch Size Determination

As previously mentioned in Chapter 2 and 3, the size of the patch directly affects the final completion accuracy since local image characteristics change through-out the image. Therefore, having a fixed patch size may result in poor completion results as shown in Figure 4.8 (b), where we use the method of [1] to complete the image using a fixed patch of size of 11x11. Intuitively, we should have small patch sizes in areas that cover many distinct objects and larger patch sizes in areas with few distinct objects as shown in Figure 4.8 (c), the completion result is shown in Figure 4.8 (d). Building upon [38] and in a manner similar to [31], we formulate the problem as an optimization problem, minimizing an objective function of local edge orientated histograms. We proceed in the following steps in the patch size determination.



(a) Image and hole (blue)

(b) Result using patch size 11x11

(c) Adaptive Patch Sizes

(d) Adaptive Patch Result

Figure 4.8: Patch Size Determination

1) Gradient Computation

Since local object appearance and shape can be characterized well by the distribution of local intensity gradients and edge directions [38], the first step of calculation is the computation of the gradient values.

We apply the 1D centered point discrete derivative mask in both the horizontal (D_x) and vertical (D_y) directions, filtering each colour channel with the following filter kernels:

$$D_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad and \quad D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}. \tag{4.5}$$

Given a colour image I we then obtain the x and y derivatives for each RGB channel using the convolution operation:

$$I_x = I * D_x \quad and \quad I_y = I * D_y, \tag{4.6}$$

and further obtain the magnitude($|G|$) and orientation(θ) of the gradients for each RGB channel calculated as

$$|G| = \sqrt{I_x^2 + I_y^2} \quad and \quad \theta = arctan\frac{I_y}{I_x}. \tag{4.7}$$

Figure 4.9 shows the magnitude and orientation for each RGB channel of a colour image.

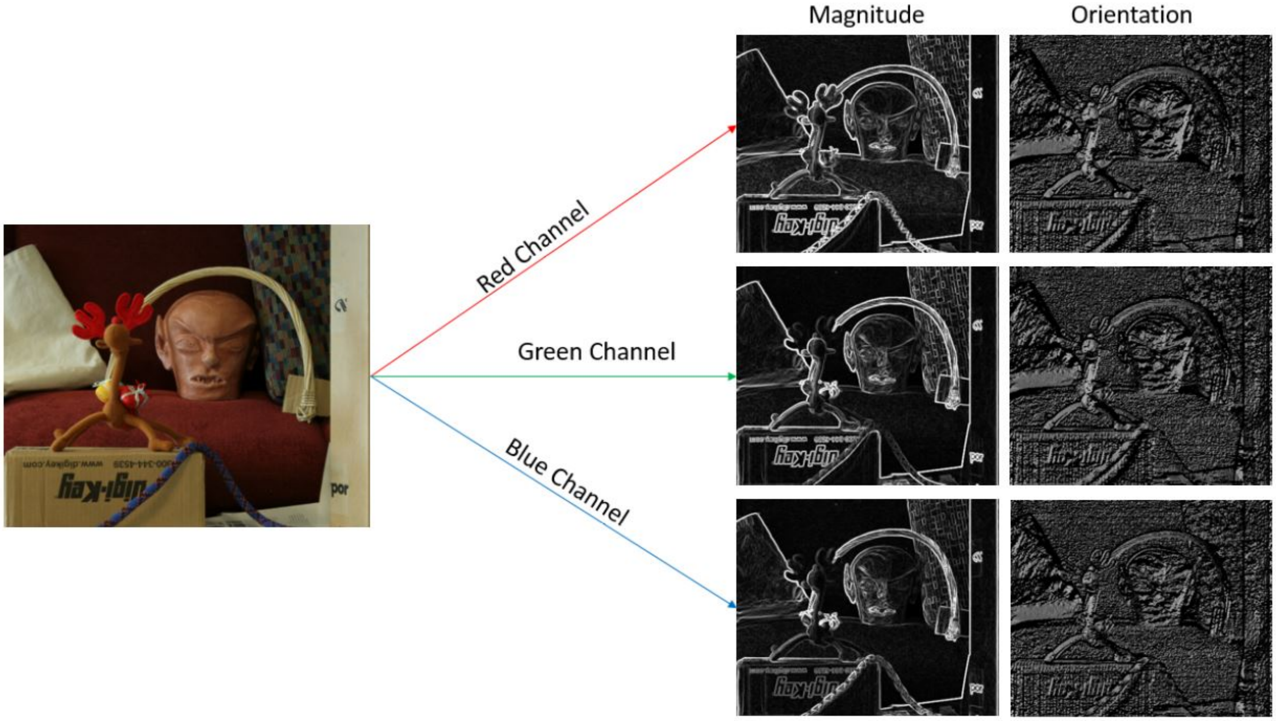


Figure 4.9: Magitude and Orientation of RGB Image

2) Orientation Binning

The second step of the calculation involves creating the edge orientated histograms. Within a square patch ($\psi_{sq}(k,p)$) of size kxk and centered at p with $p \in \Omega$, each pixel within $\psi_{sq}(k,p)$ casts a weighted vote for the histogram itself. We use 9 orientation bins that are evenly spread over 0 to 360 degrees, and each weighted vote is a function of the gradient magnitude itself at that pixel. For each pixel within $\psi_{sq}(k,p)$, we use the colour channel with the highest gradient magnitude for that pixel. Figure 4.10 shows the histograms for a subsection of an image. In Figure 4.10, it can be seen that in areas that contain a lot of edges (i.e. the wooden arch) have a corresponding histogram which contains a lot more information as opposed to areas containing no edges, this is due to gradient calculation in detecting this edges at all different angles. In Figure 4.10 we use a constant patch size of 9x9. We use the idea that these local features can be detected through the use of a histogram and develop a patch size determination in the following step.

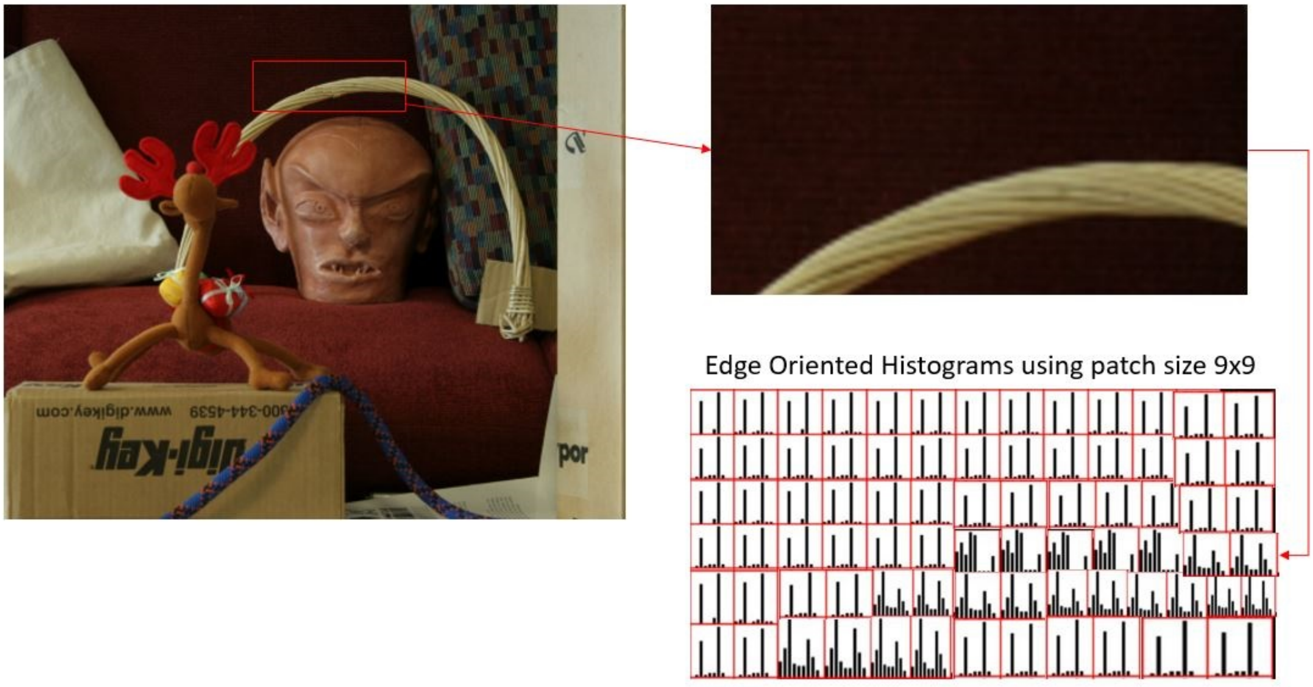


Figure 4.10: Histogram of Image Subsection

3) Determining Patch Sizes

Now for each $p \in \Omega$, we construct a sequence of edge orientated histograms over $\psi_{sq}(k, p)$ for different sizes of k and perform a similarity measure between histograms. The more differences two histograms have, the less similar the corresponding patches are, and our goal is to find an optimal patch size with the smallest structure changes inside. Let $Hist(k, p)$ be a edge orientated histogram for a square patch of size kxk and centered on pixel p, with $p \in \Omega$. Further let, $Hist(k+2, p)$ be an edge orientated histogram for a square patch of size $(k+2)x(k+2)$ and centered on pixel p, with $p \in \Omega$. For each histogram, let $P = (p_1, p_2, ..., p_9)$ be the 9 orientation bins corresponding to $Hist(k, p)$ and let $Q = (q_1, q_2, ..., q_9)$ be the 9 orientation bins corresponding to $Hist(k+2, p)$. We suggest that the optimal patch size for all $p \in \Omega$ be the solution to the following problem:

$$min(\sum_{p \in \Omega}(sim(Hist(k, p), Hist(k+2, p))), \tag{4.8}$$

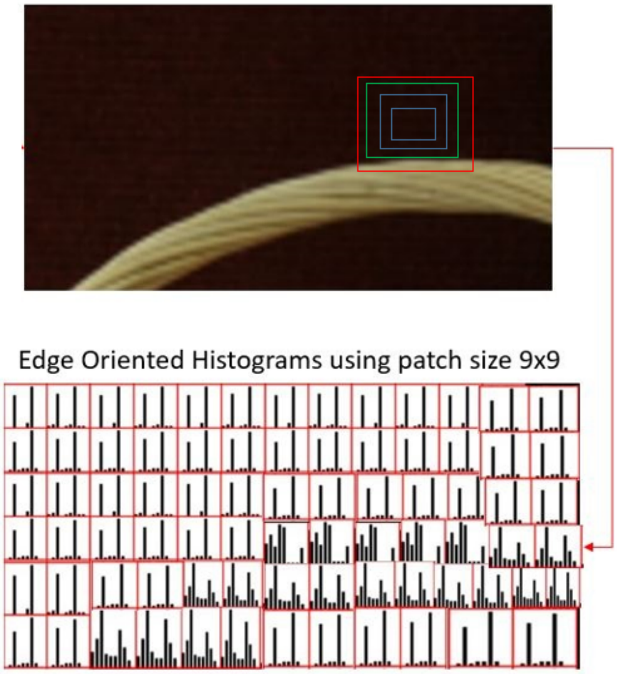


Figure 4.11: Patch Size Determination

where $sim(Hist(k,p), Hist(k+2,p)$ is the cosine similarity between the two histograms defined as:

$$sim(Hist(k,p), Hist(k+2,p)) = \frac{p_1q_1 + p_1q_1 + ... + p_9q_9}{\sqrt{p_1^2 + p_2^2 + ... + p_9^2} \cdot \sqrt{q_1^2 + q_2^2 + ... + q_9^2}} \tag{4.9}$$

The minimum value of k used in our experiments is $k = 7$ to ensure a large distribution of pixels in our histogram calculation, and we set no limit on the maximum value that patch size could obtain. Figure 4.11 demonstrates this patch size determination. The smallest blue patch is our initial 7x7 patch and we grow to 9x9 patch shown by the second slightly larger blue patch. Since texture information is the same for both patches, performing a similarity measure between there corresponding edge orientated histograms results in a large similarity value. Growing to a patch size of 11x11 shown as the green patch, we again perform a similarity measure between the 9x9 and the 11x11 patch. Since texture information is the same the resulting similarity value is high. Growing once again to a 13x13 patch shown as the red patch we perform a similarity measure better the 11x11 and the 13x13 patch. However, in the red patch, new edge information is now introduced which affects the edge oriented histogram. Performing a similarity measure between the 13x13 and the 11x11 patches results in a low similarity. Therefore we found a minimum value and we would use the green patch as our determined patch. The minimum in equation (4.8) searches for the first occurrence of this dissimilarity between patches. The patch size starts at a size of 7x7 and will continue to grow until a dissimilarity is found between edge oriented histograms.

4.2.4 Propagation

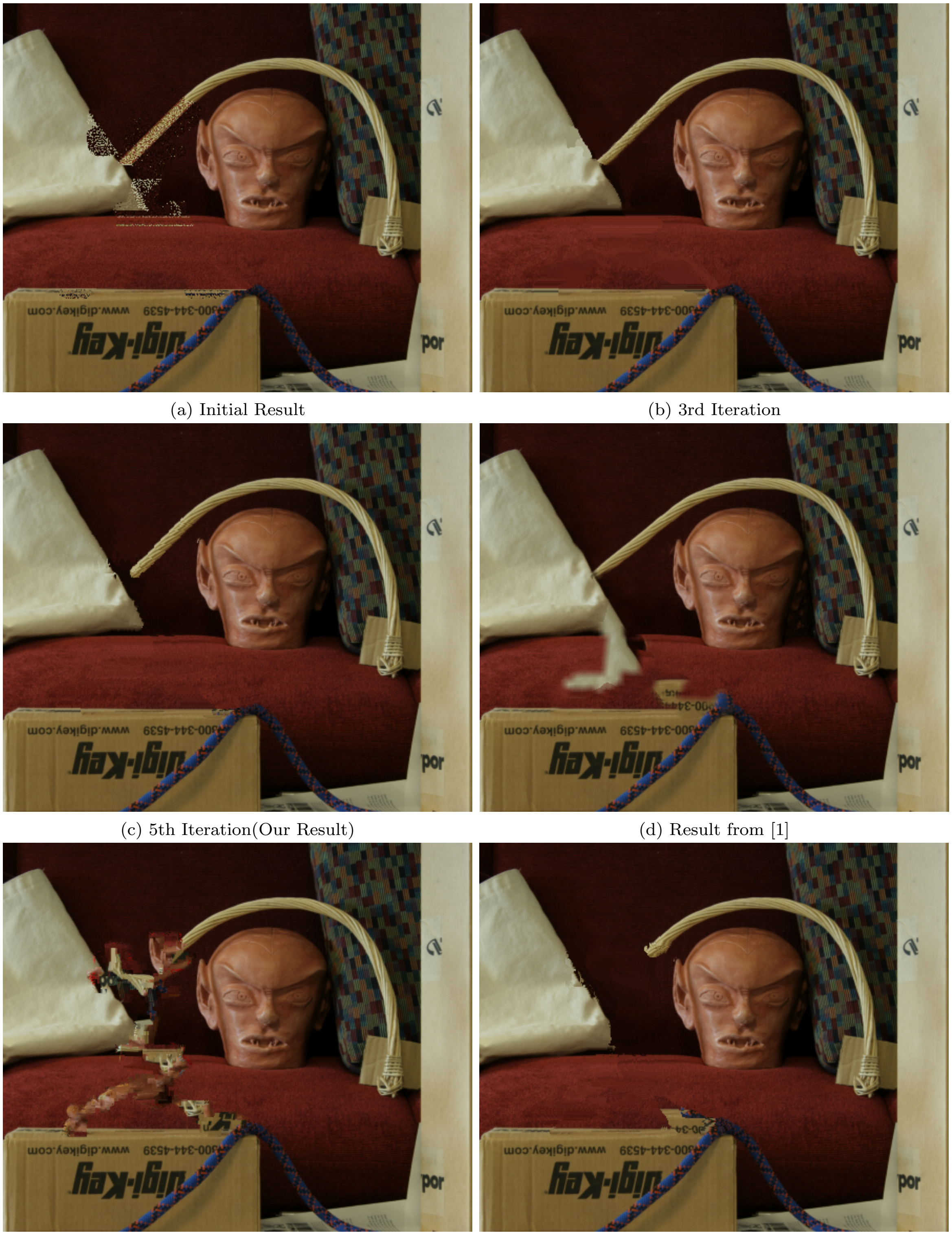
After the initialization step and once the patch size is determined, then for each overlapping patch ψ_p for every $p \in \Omega$ we compute the offset to the nearest neighbour patch ψ_q with $q \in \Psi$. We perform the same propagation steps as discussed in section 3.2.1 by checking neighbouring offsets on odd and even iterations.

4.2.5 Random Search

As depth control reduces the search space in the initialization step, the first 3 iterations we do not perform random searching. This is to avoid falling in suboptimal local minima. In subsequent iterations, we perform random searching as explained in section 3.2.1.

4.3 Colour and Depth Updating

For each unknown pixel, we update our colour and depth information using the weighting proposed in section 3.2.1 and used by [1]. Figure 4.12 and 4.13 (a) (b) (c) shows our results after Initial, third and fifth iterations for both the colour and depth completion, respectively. Figure 4.12 and 4.13 (d) (e) (f) shows both the colour and depth completion completion results for [1], [23] and [10], respectively.



(a) Initial Result

(b) 3rd Iteration

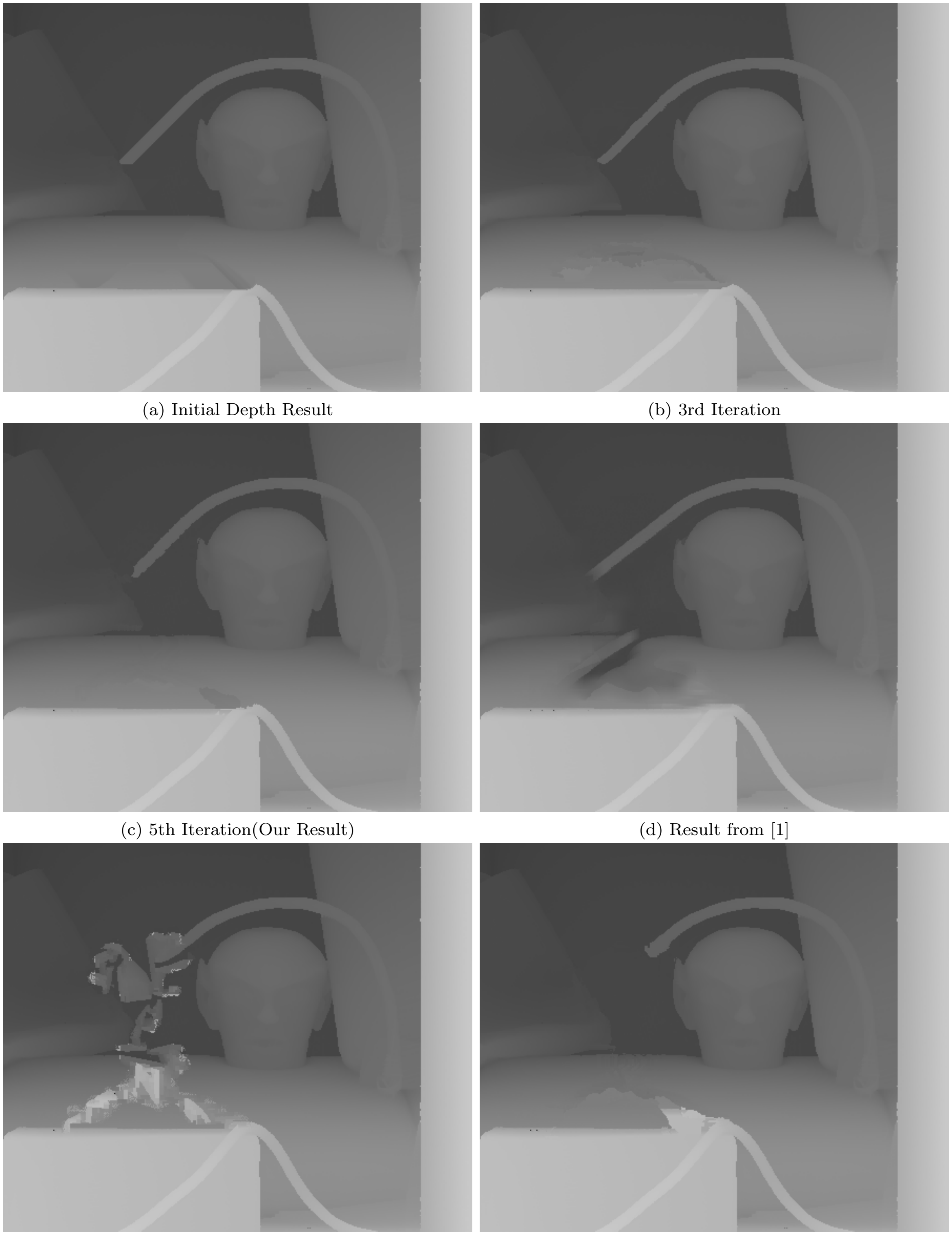
(c) 5th Iteration(Our Result)

(d) Result from [1]

(e) Result from [23]

(f) Result from [10]

Figure 4.12: Colour Image Completion Results



(a) Initial Depth Result

(b) 3rd Iteration

(c) 5th Iteration(Our Result)

(d) Result from [1]

(e) Result from [23]

(f) Result from [10]

Figure 4.13: Corresponding Depth Completion Results

Chapter 5

Results and Comparsions

We test our algorithm on Middlebury datasets [39,40] and datasets provided by [41,42] where the colour images and corresponding depth maps are obtained through video sequences. Figures 5.1-5.4 and Figures 5.9-5.11 show test results produced by no-depth guiding [1, 23], depth-guided inpainting [10] and our filling scheme discussed, i.e. Random initialization, adaptive patch determination, iterative propagation and random search. On the images inpainted without depth constraints, textural content is drawn from inappropriate background exemplars resulting in undesired structural completion as shown in figures 5.1-5.4 c,d,e. Such errors are avoided in our results with our algorithm since we depth constrain our exemplar search to only consider exemplars constrained to the completed depth information. We further show that our method produces overall improved results for object removal as shown in figures 5.9-5.11. Specifically, if we examine figure 5.10, the results of figure 5.10 c,d,e produce unwanted artifacts, failure of linking similar regions across the hole and error propagation, respectively. Our result in figure 5.10 (f) avoids these unsatisfying results by using depth information to correctly link similar regions across the hole and an improved SSD criteria to avoid unwanted artifacts and error propagation.Since methods [1, 23] do not perform depth map completion, we only compare our colour image results. Our depth completion results we directly compare with the method of [10] since they perform simultaneous colour image and depth completion. The results of He et al. [10] constrain the search space to only consider farther depth values, which leads to unsatisfying results when the hole contains structural information as shown in figures 5.1-5.4 since it is only appropriate to search behind the object only for object removal.

To further conduct comparisons, we take the images in figure 5.1-5.4 (a) as groundtruth, and randomly remove some region in these images as shown in red in figure 5.1-5.4 (b). We then run the completion algorithms of [1,10,23] and our proposed method to complete the randomly removed region. For objective evaluation, we use the peak signal-to-noise ratio (PSNR) to measure the completion quality given as:

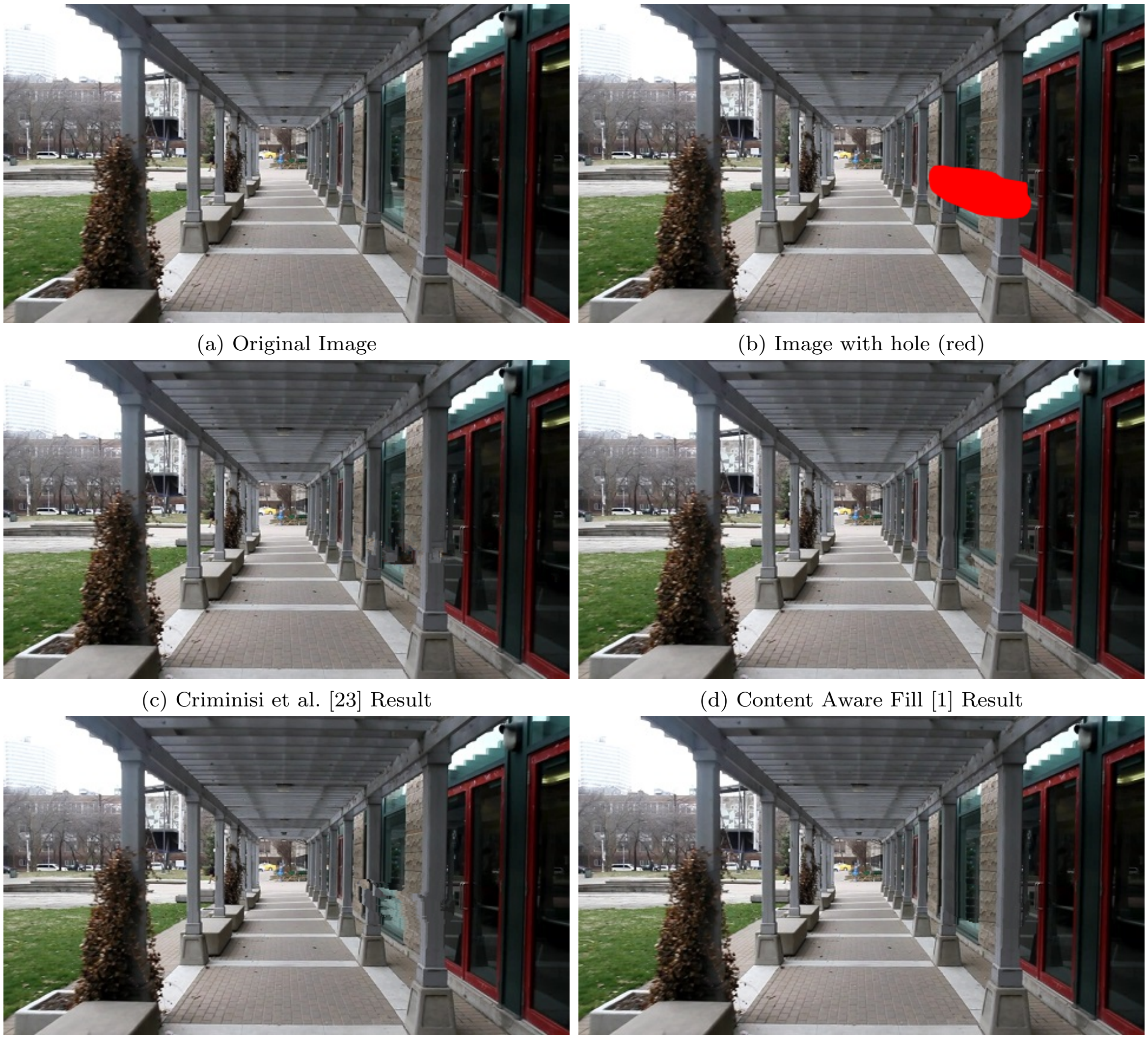
$$PSNR = 10log_{10}(\frac{255}{\frac{1}{M}\sum_{p\in T}|I_o(p)-I_c(p)|}), \tag{5.1}$$

Where $I_o(p)$ and $I_c(p)$ are the values at pixel p in the original and completed images respectively,$|I_o(p)-I_c(p)|$ is the absolute differences in the RGB colour space, T is the target region and M is the number of pixels in the target region. Table 5.1 shows results produced by our approach and other algorithms. It can been seen in table 5.1 that our approach produces better or comparable results then the previous methods. However, since the PSNR objective measure only attempts to quantify the visibility of errors between the inpainted image and the original image, any colour mismatch between images will result in a lower PSNR which implies a less satisfying completion result. Image completion aims in producing a coherent and visually plausible result and the optimal solution may differ from the original image resulting in poor signal-to-noise ratio (PSNR) but having high visual quality. Therefore, we further use a structural similarity index measurement (SSIM) introduced in [43] for assessing the image quality. Since human visual perception is highly adapted for extracting structural information, the quality assessment is based on the degradation of this structural information with 0 being poor quality assessment and 1 being the highest. Table 5.2 shows the results for our method and previous approaches.Examining the results of table 5.2 we see that the SSIM may indicate that the results are of similar quality, e.g., Figure 5.1, we see 0.9888, 0.9884, 0.9857 and 0.9896, since the quality assessment for each method contains minor differences. Also, Figure 5.2 has a higher SSIM value for previous methods than the proposed method indicating previous approaches outperform our proposed method.Therefore, to further demonstrate the effectiveness of our approach, we conducted a perceptual survey in which 100 randomly selected human volunteers were asked to select which image produced the most satisfying result and our findings are tabulated in table 5.3. As can be seen in table 5.3, our results had the highest percentage of human selection and it was clearly the preferred result from among the 100 human volunteers. While the SSIM indicates that many results may be of equal image quality and in some cases fail to demonstrate the effectiveness of our proposed method. Examining the human perceptual results we see that this is not the case but that our proposed method is the preferred result for each case. While figure 5.2 has a higher SSIM value for Criminisis method than the proposed method, human perceptual results show that our method exhibits much higher perceptual quality as it was selected 79.6% while the Criminisi only 12.6%. In all cases, the perceptual survey shows the effectiveness of our adaptive depth guided approach as opposed to non-depth guided results.

5.1 Qualitative Results

5.1.1 Colour Image Completion Results for Random Removal

In the followings figures, we randomly remove a section of the image and complete the missing parts. We compare our method to [1, 10, 23].



(a) Original Image

(b) Image with hole (red)

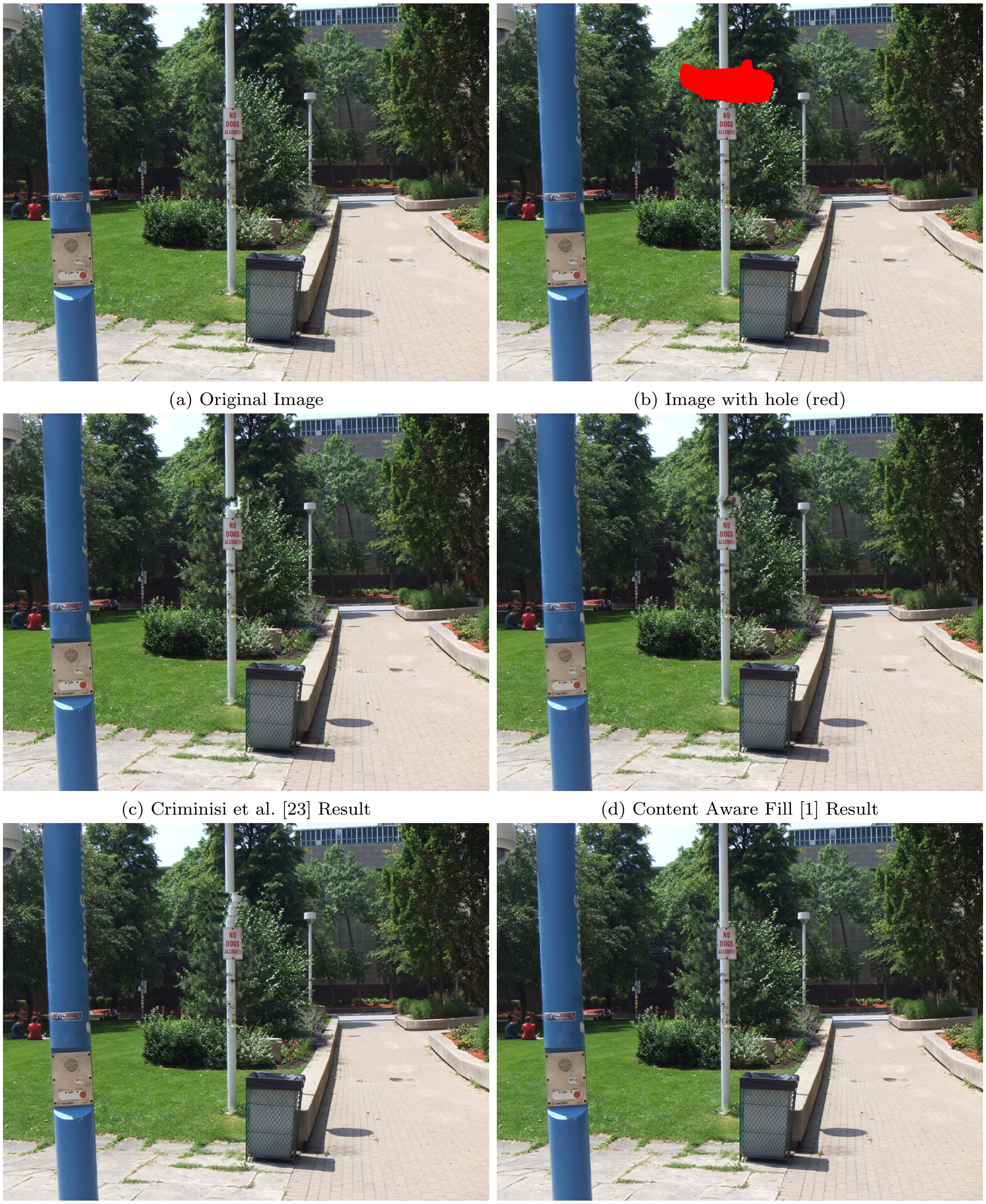
(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

(e) He et al. [10] Result

(f) Our Result

Figure 5.1: Colour Image Completion Results - Ryerson Pathway



(a) Original Image

(b) Image with hole (red)

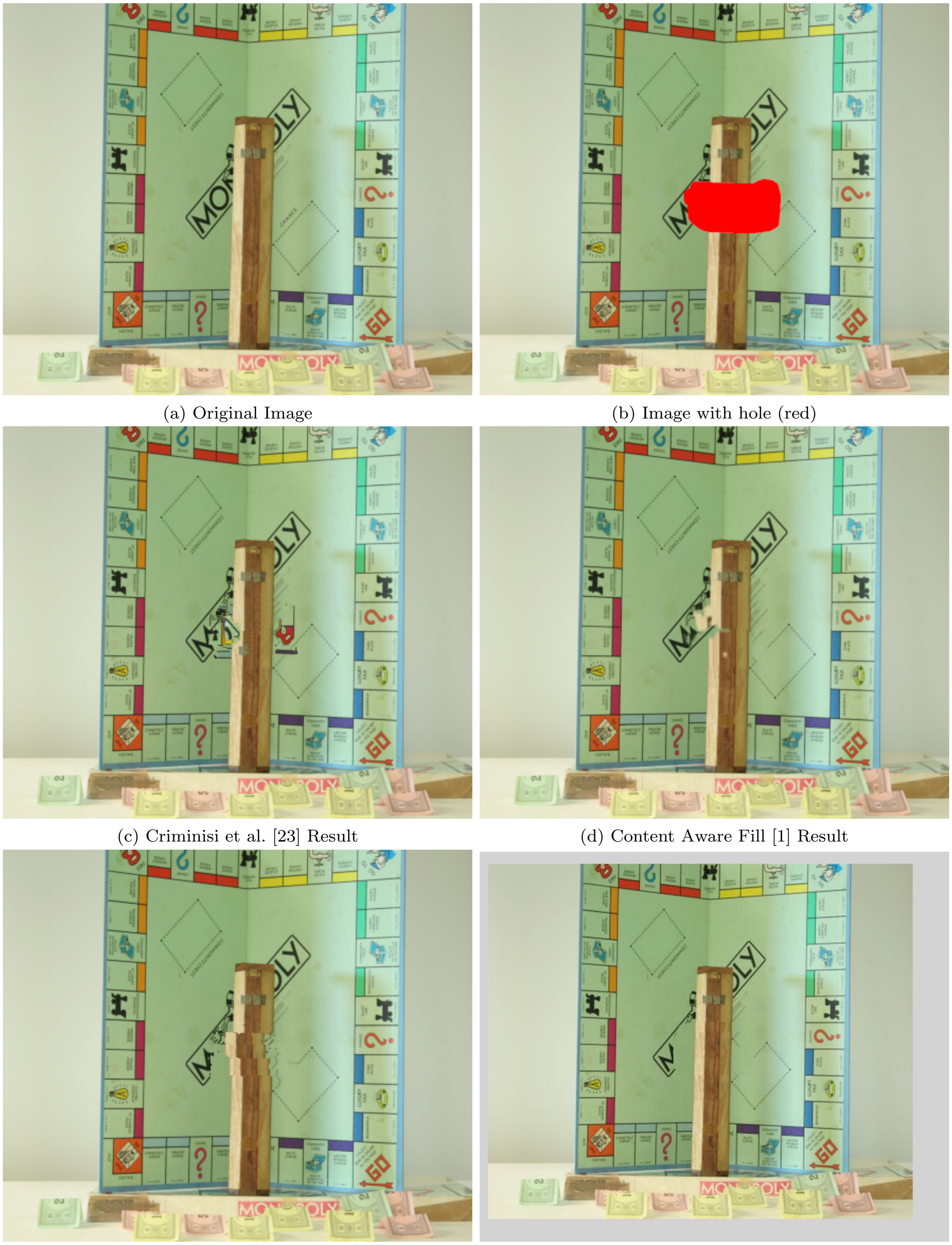
(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

(e) He et al. [10] Result

(f) Our Result

Figure 5.2: Colour Image Completion Results - Ryerson Yard



(a) Original Image

(b) Image with hole (red)

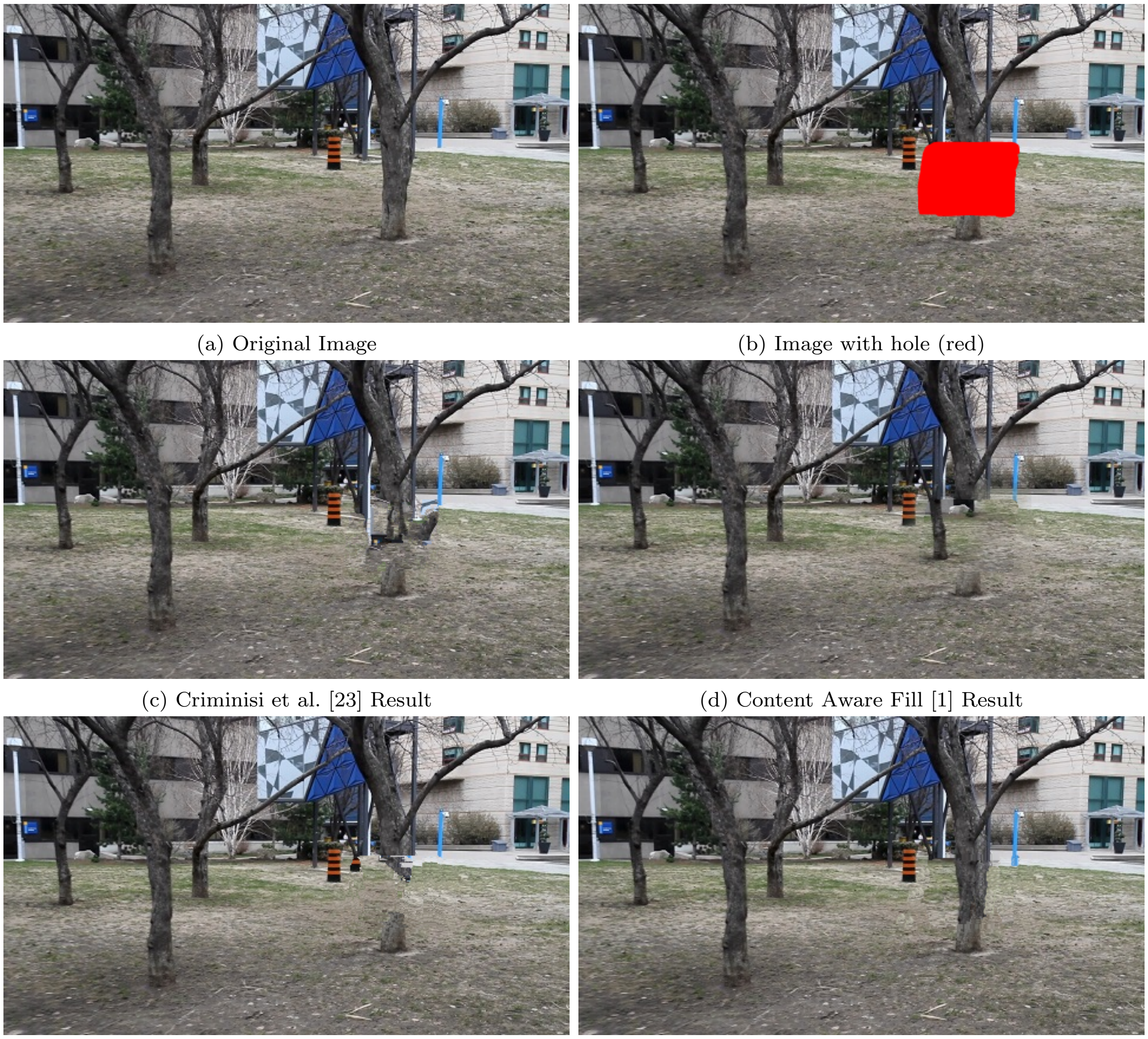
(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

(e) He et al. [10] Result

(f) Our Result

Figure 5.3: Colour Image Completion Results - Monopoly



(a) Original Image

(b) Image with hole (red)

(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

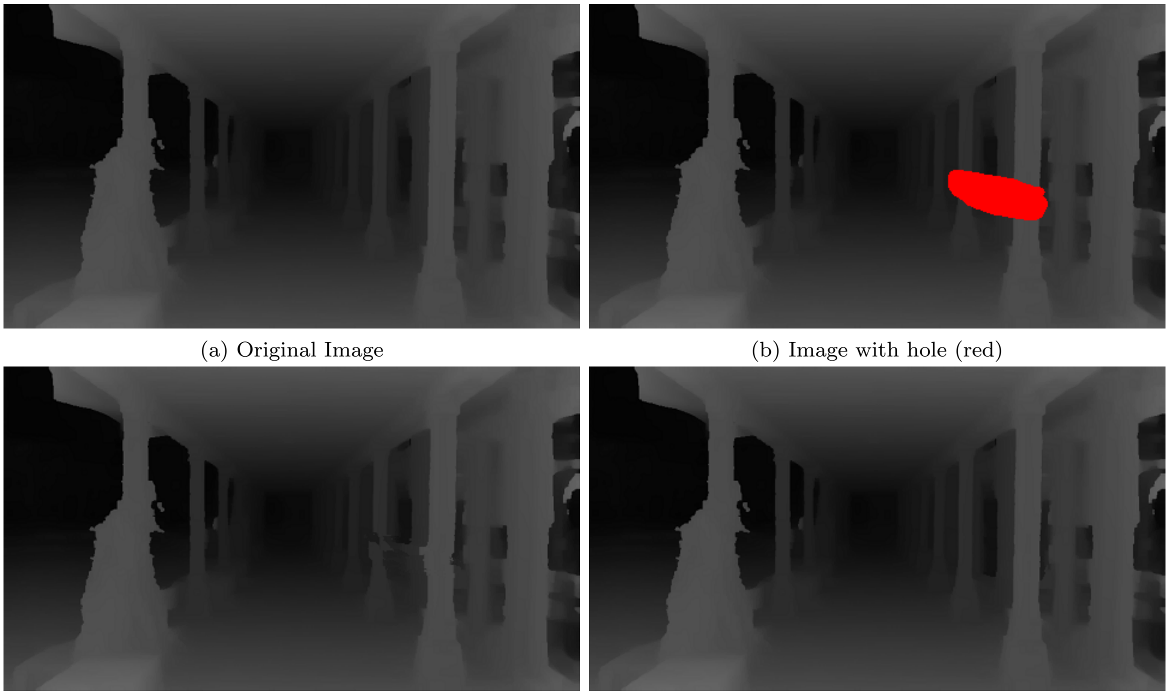
(e) He et al. [10] Result

(f) Our Result

Figure 5.4: Colour Image Completion Results - Ryerson Campus

5.1.2 Corresponding Depth Completion for Random Removal

The following figures are the corresponding depth completion for figures 5.1,5.2,5.3,5.4, respectively. As previously noted, we only compare our depth completion results to He. et al [10], since they preform simultaneous colour and depth completion.



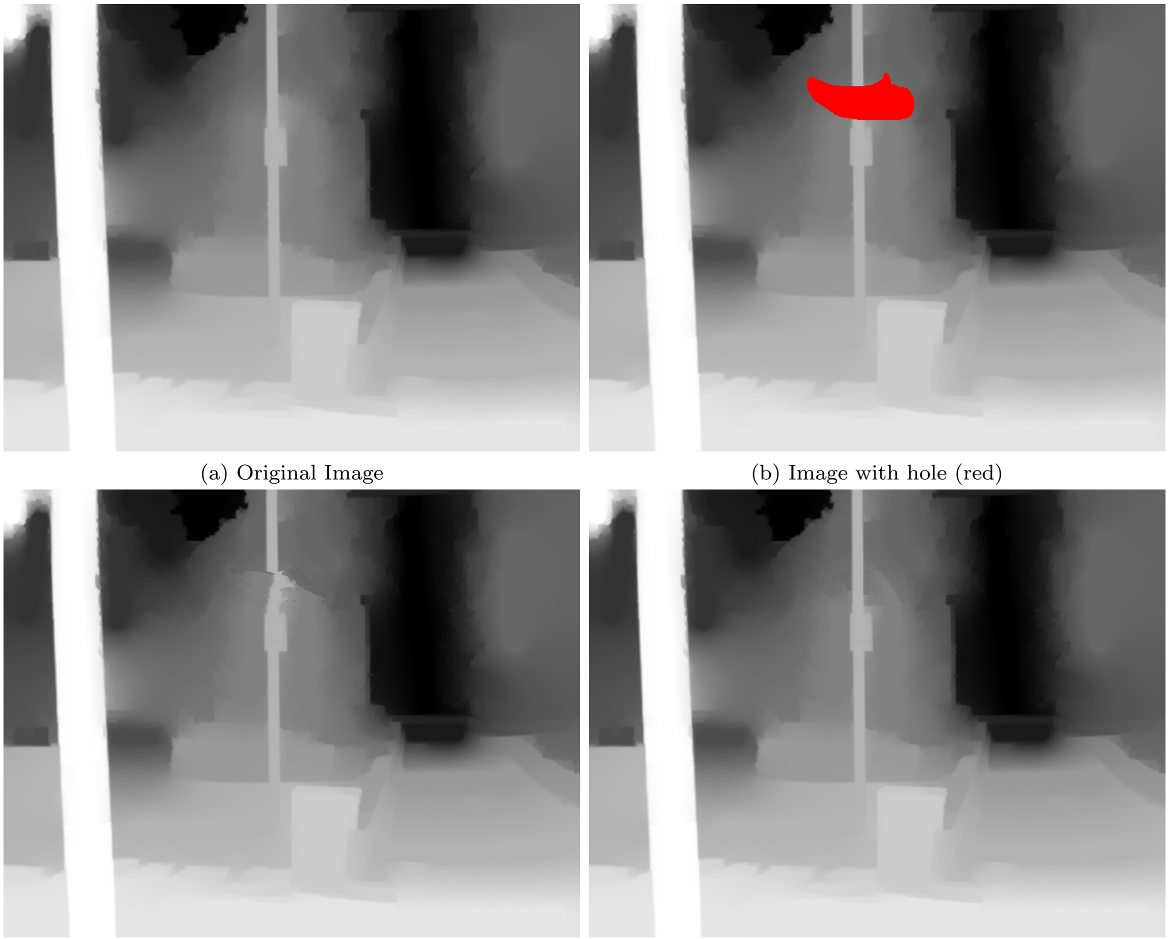
(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Result

(d) Our Result

Figure 5.5: Depth Completion Results - Ryerson Pathway



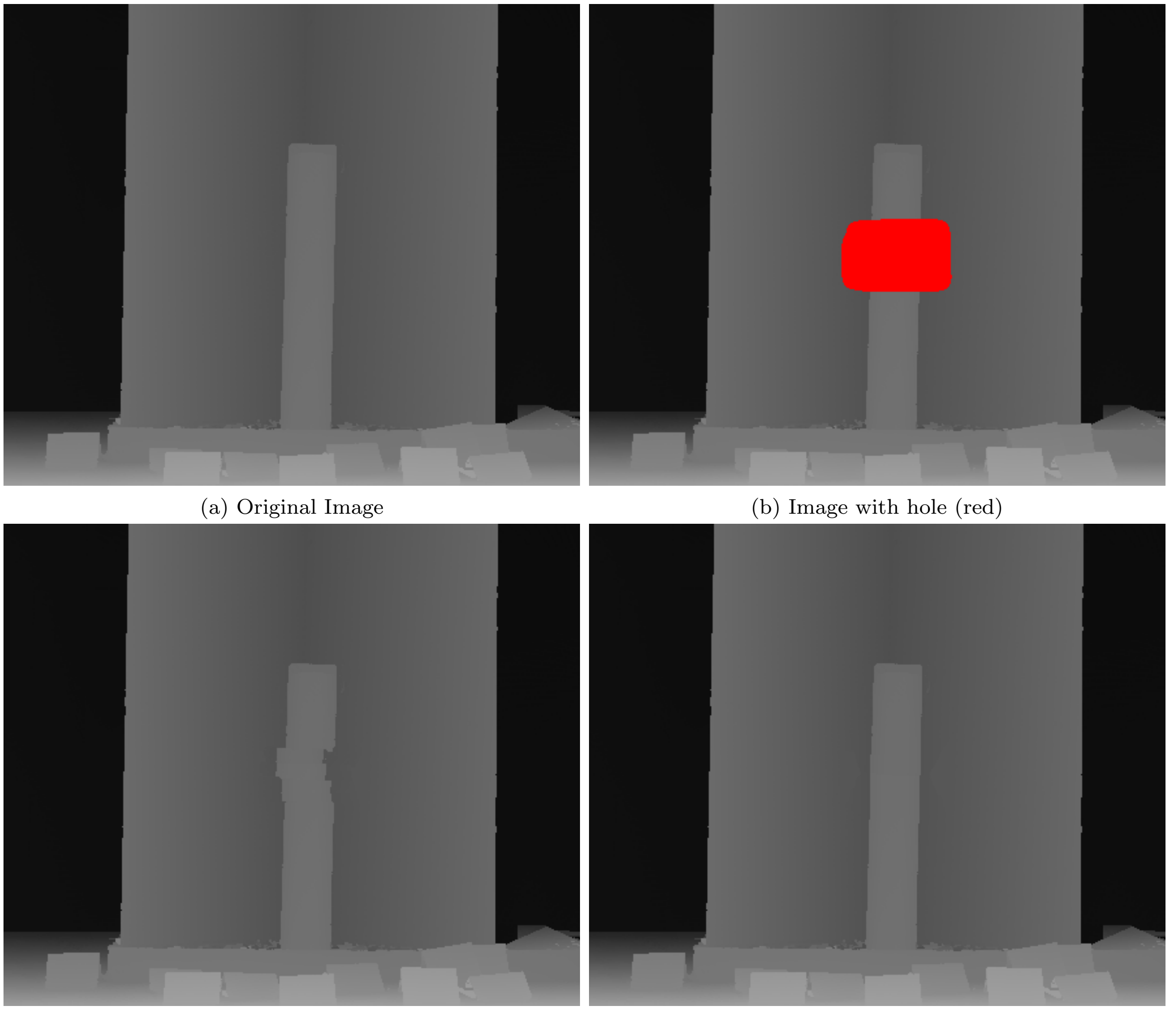
(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Result

(d) Our Result

Figure 5.6: Depth Completion Results - Ryerson Yard



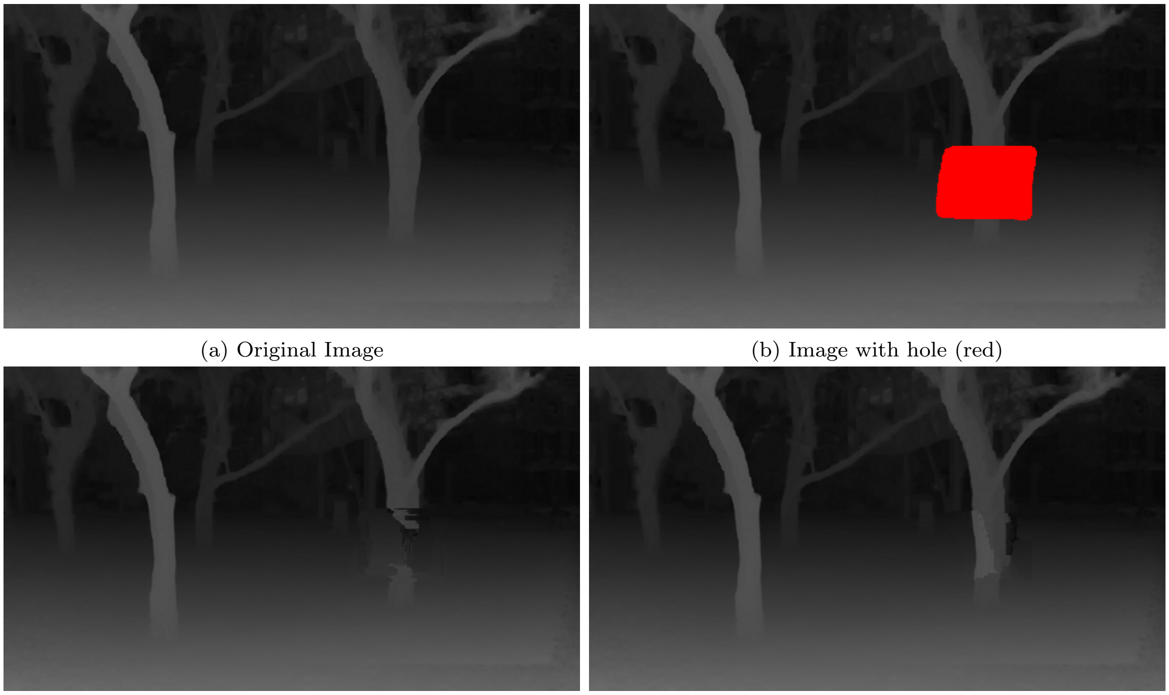
(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Result

(d) Our Result

Figure 5.7: Depth Completion Results - Monopoly



(a) Original Image

(b) Image with hole (red)

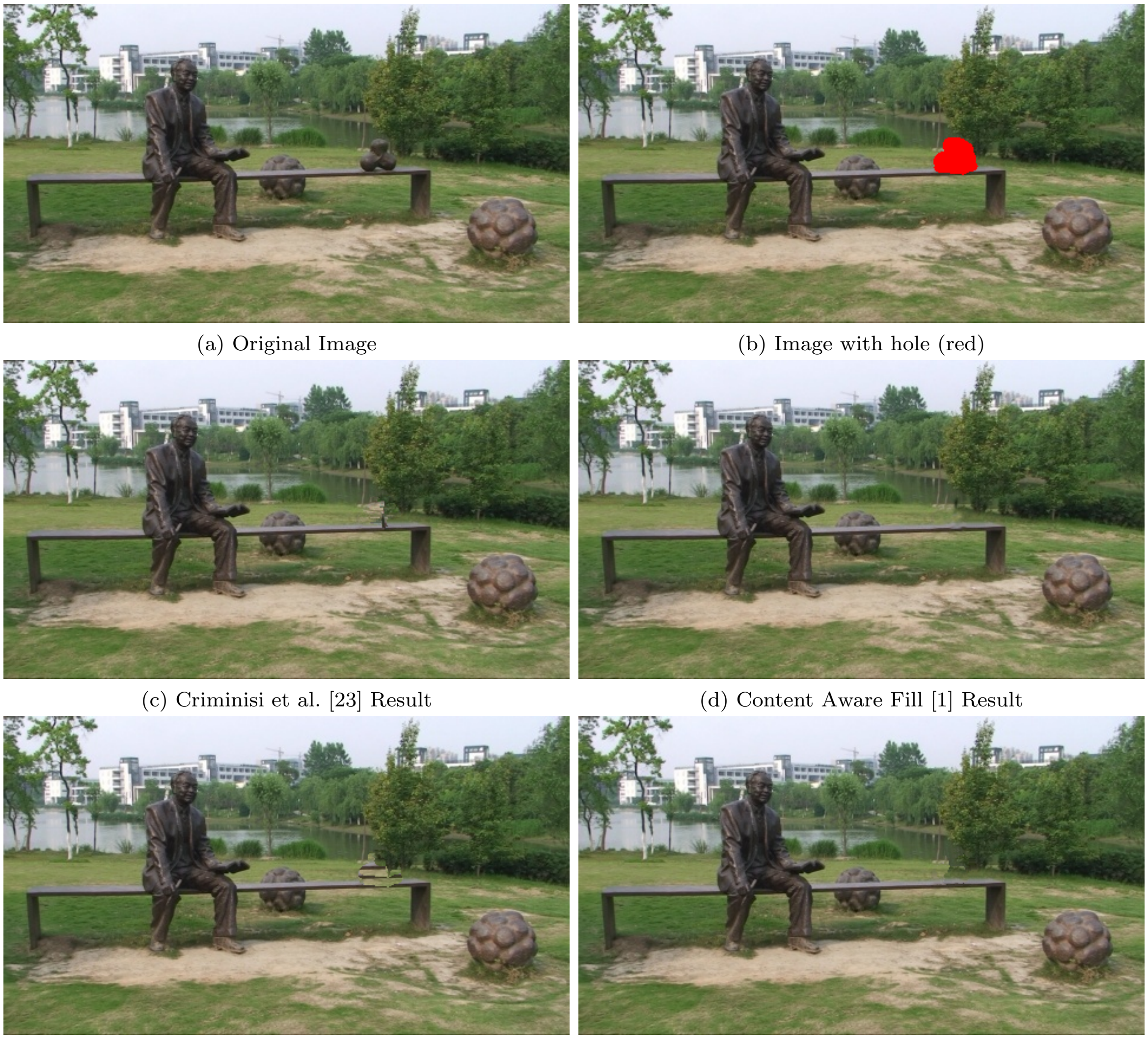
(c) He et al. [10] Result

(d) Our Result

Figure 5.8: Depth Completion Results - Ryerson Campus

5.1.3 Colour Image Completion Results for Object Removal

In the followings figures, we randomly remove an object in the image and complete the missing parts. We again compare our method to [1, 10, 23].



(a) Original Image

(b) Image with hole (red)

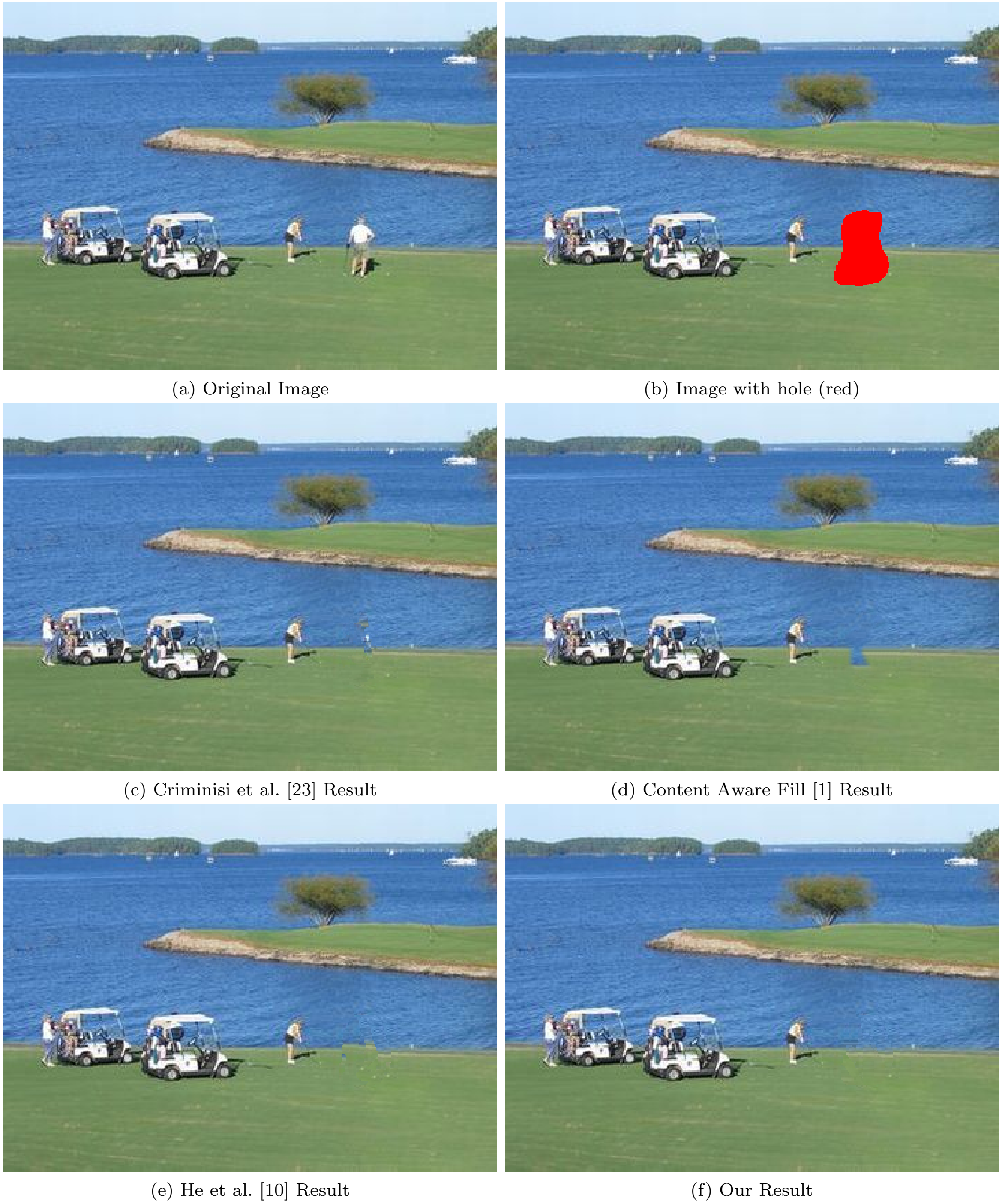
(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

(e) He et al. [10] Result

(f) Our Result

Figure 5.9: Object Removal Completion Results - Lawn



(a) Original Image

(b) Image with hole (red)

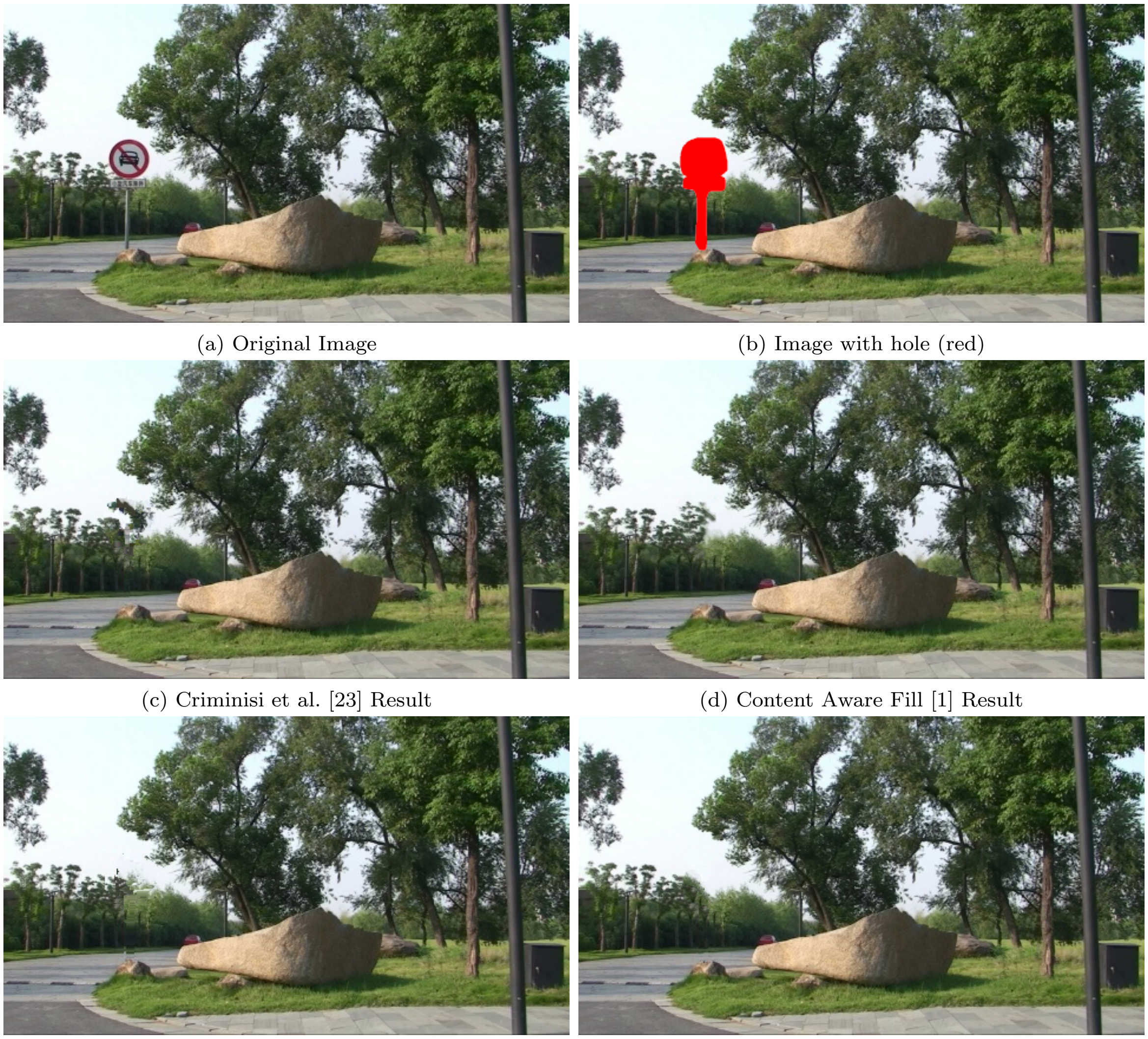
(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

(e) He et al. [10] Result

(f) Our Result

Figure 5.10: Object Removal Completion Results - Golf



(a) Original Image

(b) Image with hole (red)

(c) Criminisi et al. [23] Result

(d) Content Aware Fill [1] Result

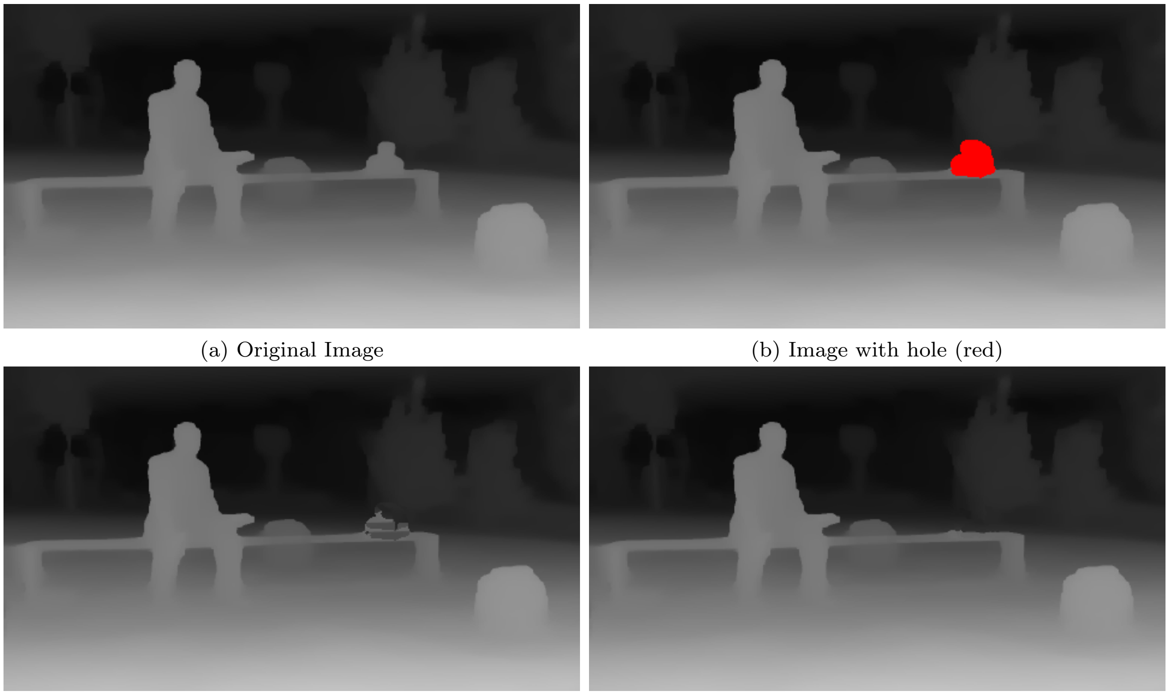
(e) He et al. [10] Result

(f) Our Result

Figure 5.11: Object Removal Completion Results - Road

5.1.4 Corresponding Depth Completion for Object Removal

The following figures are the corresponding depth completion for figures 5.9,5.10,5.11, respectively. As previously noted, we only compare our depth completion results to He. et al [10], since they preform simultaneous colour and depth completion.



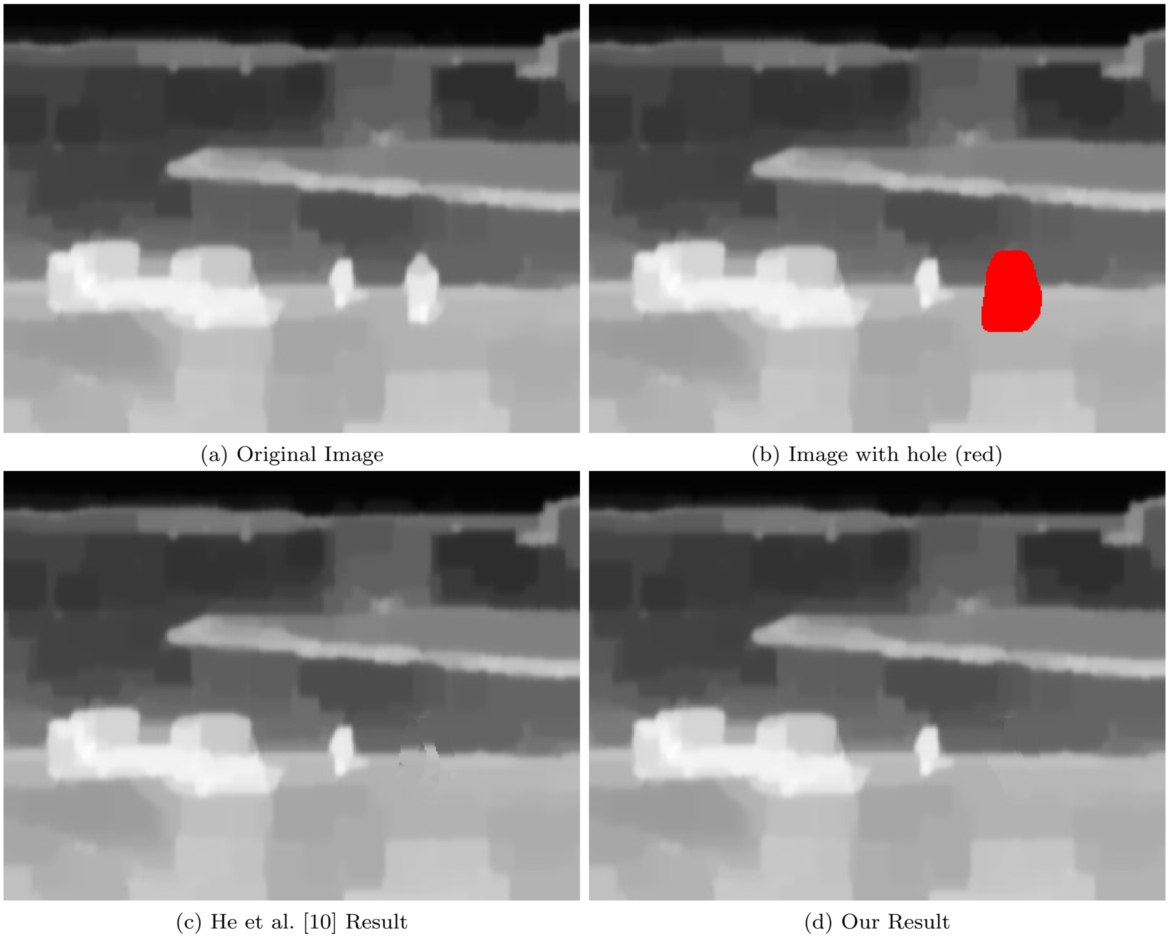
(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Result

(d) Our Result

Figure 5.12: Object Removal Depth Completion Results - Lawn



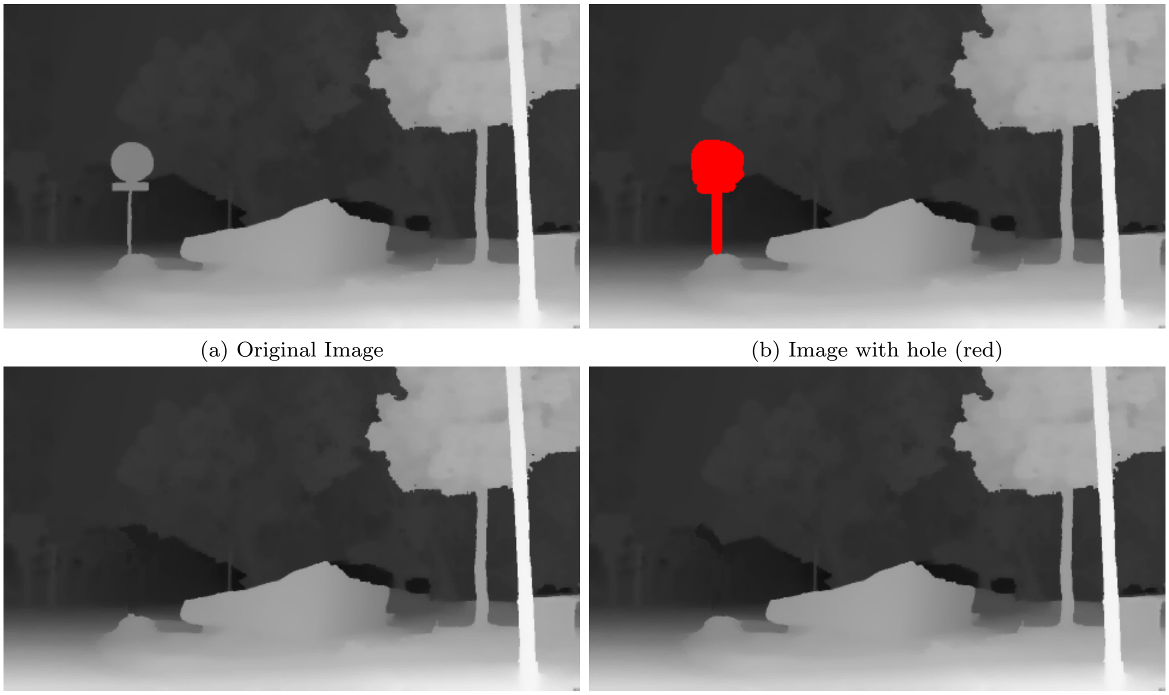
(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Result

(d) Our Result

Figure 5.13: Object Removal Depth Completion Results - Golf



(a) Original Image

(b) Image with hole (red)

(c) He et al. [10] Result

(d) Our Result

Figure 5.14: Object Removal Depth Completion Results - Road

5.2 Quantitative Results

Image	**PSNR (dB)**			
	Criminsi et al.	**Content Aware Fill**	**He et al. Method**	**Proposed Method**
Figure 5.1 - Pathway	30.0585	31.1074	29.0667	31.1442
Figure 5.2 - Ryerson Yard	27.5789	27.7083	27.3805	28.5336
Figure 5.3 - Monopoly	29.6622	30.0281	29.6762	31.6055
Figure 5.4 - Ryerson Campus	27.3945	27.0454	27.8492	29.2307

Table 5.1: PSNR Calculations

Image	**SSIM Index**			
	Criminsi et al.	**Content Aware Fill**	**He et al. Method**	**Proposed Method**
Figure 5.1 - Pathway	0.9888	0.9884	0.9857	0.9896
Figure 5.2 - Ryerson Yard	0.9892	0.9890	0.9883	0.9654
Figure 5.3 - Monopoly	0.9139	0.9590	0.9253	0.9854
Figure 5.4 - Ryerson Campus	0.9639	0.9590	0.9653	0.9654

Table 5.2: SSIM Index Calculations

Image	**Selected Image (/100)**			
	Criminsi et al.	**Content Aware Fill**	**He et al. Method**	**Proposed Method**
Figure 5.1 - Pathway	10.7%	3.9%	2.9%	82.5%
Figure 5.2 - Ryerson Yard	4.9%	7.8%	3.9%	83.5%
Figure 5.3 - Monopoly	12.6%	6.8%	1%	79.6%
Figure 5.4 - Ryerson Campus	4.9%	6.8%	1%	87.4%
Figure 5.10 - Golf	18.4%	2.9%	18.4%	77.7%
Figure 5.11 - Road	5.8%	28.2%	6.8%	59.2%

Table 5.3: Subjective Survey Results

5.3 Limitation of Proposed Method

For the colour image results shown in Figures 5.1-5.4 and 5.9-5.11, the proposed method performs well when the objects are fronto-parallel to the camera (i.e. have defined depth) as shown in Figures 5.5-5.8 and 5.12-5.14. However, for cases when objects contain varying depth information (i.e. going into the page) the proposed method fails when a hole falls within this region. This is because in the depth completion, directly copying and pasting pixels is inappropriate since depth in this case is slowly varying.

Chapter 6

Conclusion and Future Work

In this thesis project, we have presented a method that fills in a hole left by the removal of an unwanted or undesired area in an image. In contrast to other single image completion methods, our approach considers the structure propagation in allowing similar regions separated by the hole to be linked in the resulting depth map. Our method first creates this structure by completing its corresponding depth map through the use of a diffusion based completion process. This completed depth information is then used as a constraint to guide structure and texture synthesis which is achieved using an exemplar-based method of [9] for the corresponding colour image completion. We extend the objective function of the exemplar-based method by incorporating a variance weighting, avoiding error accumulation and propagation during the exemplar searching. Further, to avoid the ambiguity of the patch size determination, we adaptively determine an optimal patch size based upon local image characterises. This requires no user input in the completion of both the colour and depth images. This method can create structure that is realistic at all scales and texture that shares visual coherence with its surrounding information as shown in Figures 5.1-5.4 and Figures 5.9-5.11. To further conduct comparisons, we calculate the PSNR and SSIM index used to demonstrate how our method compares quantitatively to previous approaches as shown table 4.1 and 4.2 respectively. While the PSNR and SSIM index values may not show our methods superiority, we conduct a subject survey shown in table 4.3, which clearly shows the effectiveness of our approach in comparison to previous approaches.

6.1 Future Work

Firstly, one possible extension would be to allow completion of objects that contain varying depth information (i.e. the limitation). This could be achieved by using another domain that is sensitive to small changes in information such as a gradient domain.

Future projects may also consist of extending the ideas from this thesis project from images to videos. With 3D video devices being made widely available, a future project may consist of using this stereoscopic information for real-time video completion. Devices such as the Xbox connect, oculus rift and stereoscopic cameras are just some the devices that could potentially be used. However, before considering extending this idea for video completion, some improvements need to be made in the existing algorithm. Since 3D information comes in pairs (i.e. Right and Left view), the first step for video completion would be to allow for stereoscopic consistency in the completion process. This means allowing the existing right view information to be used to complete the left view and vice versa. Also, since video contains additional information (i.e. Frames), an extension to the optimization equation needs to be employed to consider the consistency between frames. The end idea should be an algorithm that allows for real-time completion in 3D video.

References

[1] C. Barnes, E. Shechtman, I. Belaunde, D.B Goldman, and J. Chien, "Adobe content-aware fill," http://www.adobe.com/technology/projects/content-aware-fill.html, 2015, [Online; accessed 19-May-2015].

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

[3] N. Alotaibi and F. Labrosse, "Image completion by structure reconstruction and texture synthesis," *Pattern Analysis and Applications*, vol. 18, no. 2, pp. 333–350, 2015.

[4] S. Zarif, I. Faye, and D. Rohaya, "A comparative study of different image completion techniques," in *Computer and Information Sciences (ICCOINS), 2014 International Conference on*, June 2014, pp. 1–6.

[5] F. Wu X. Sun, Z. Xiong and S. Li, "Image compression based on parameter-assisted inpainting," United States Patent 8,774,531, July 2014.

[6] H. Hirose, "Mobile authentication by image inpainting," United States Patent 8,887,255, November 2014.

[7] Y. Tian, "Color document image segmentation and binarization using automatic inpainting," United States Patent 9,042,649, May 2015.

[8] A. Desolneux, L. Moisan, and JM. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, Springer Publishing Company, Incorporated, 1st edition, 2007.

[9] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *Communications of the ACM*, vol. 28, no. 3, pp. 24:1–24:11, July 2009.

[10] L. He, M. Bleyer, and M. Gelautz, "Object removal by depth-guided inpainting," in *ÖAGM / AAPR Workshop 2011*, (2011), pp. 1–8.

[11] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2000, SIGGRAPH '00, pp. 417–424, ACM Press/Addison-Wesley Publishing Co.

[12] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *Image Processing, IEEE Transactions on*, vol. 10, no. 8, pp. 1200–1211, Aug 2001.

[13] M. Bertalmio, A.L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I–355–I–362 vol.1.

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

[15] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436 – 449, 2001.

[16] H. Guo, N. Ono, and S. Sagayama, "A structure-synthesis image inpainting algorithm based on morphological erosion operation," in *Image and Signal Processing, 2008. CISP '08. Congress on*, May 2008, vol. 3, pp. 530–535.

[17] N. Jawas and N. Suciati, "Image inpainting using erosion and dilation operation," in *International Journal of Advanced Science and Technology*, Feburary 2013, vol. 51.

[18] A.A. Efros and T.K. Leung, "Texture synthesis by non-parametric sampling," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, vol. 2, pp. 1033–1038 vol.2.

[19] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *Image Processing, IEEE Transactions on*, vol. 16, no. 11, pp. 2649–2661, Nov 2007.

[20] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 151–158.

[21] T.S. Cho, M. Butman, S. Avidan, and W.T. Freeman, "The patch transform and its applications to image editing," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[22] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 12, pp. 2423–2435, Dec 2014.

[23] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *Trans. Img. Proc.*, vol. 13, no. 9, pp. 1200–1212, Sept. 2004.

[24] C-C. Kuo P-Y. Huang and H-C. Huang, "Systems and methods for performing image inpainting based on texture analysis," United States Patent 8,867,794, October 2014.

[25] Y.F Liu, F. Wang, and X.n Xi, "Enhanced algorithm for exemplar-based image inpainting," in *Computational Intelligence and Security (CIS), 2013 9th International Conference on*, Dec 2013, pp. 209–213.

[26] N. Komodakis, "Image completion using global optimization," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, June 2006, vol. 1, pp. 442–452.

[27] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[28] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[29] H-P. Hung and H-C. Huang, "Systems and methods for multi-resolution inpainting," United States Patent 9,014,474, April 2015.

[30] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-based image completion,” in *ACM SIGGRAPH 2003 Papers*, New York, NY, USA, 2003, SIGGRAPH ’03, pp. 303–312, ACM.

[31] H. Zhou and J. Zheng, “Adaptive patch size determination for patch-based image completion,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 421–424.

[32] N.J. Butko, L. Zhang, G.W. Cottrell, and J. R. Movellan, “Visual saliency model for robot cameras,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 2398–2403.

[33] H. Hirschmuller and D. Scharstein, “Evaluation of cost functions for stereo matching,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, June 2007, pp. 1–8.

[34] R. Phan, R. Rzeszutek, and D. Androutsos, “Semi-automatic 2d to 3d image conversion using scale-space random walks and a graph cuts based depth prior,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept 2011, pp. 865–868.

[35] S. A. Jones, *GIMP - The Official Manual: 718-pages Manual, The HOW TO to Photo Retouching, Image Composition and Image Authoring*, CreateSpace, Paramount, CA, 2008.

[36] K.J. Oh, S. Yea, and Y.S. Ho, “Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video,” in *Picture Coding Symposium, 2009. PCS 2009*, May 2009, pp. 1–4.

[37] H. Winnemller, J.E. Kyprianidis, and S.C. Olsen, “Xdog: An extended difference-of-gaussians compendium including advanced image stylization,” *Computers Graphics*, vol. 36, no. 6, pp. 740 – 753, 2012, 2011 Joint Symposium on Computational Aesthetics (CAe), Non-Photorealistic Animation and Rendering (NPAR), and Sketch-Based Interfaces and Modeling (SBIM).

[38] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, June 2005, vol. 1, pp. 886–893 vol. 1.

[39] H. Hirschmuller and D. Scharstein, “Evaluation of cost functions for stereo matching,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, June 2007, pp. 1–8.

[40] D. Scharstein, H. Hirschmller, Y. Kitajima, G. Krathwohl, N. Nei, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Pattern Recognition*, Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, Eds., vol. 8753 of *Lecture Notes in Computer Science*, pp. 31–42. Springer International Publishing, 2014.

[41] R. Rzeszutek and D. Androutsos, "A framework for estimating relative depth in video," *Computer Vision and Image Understanding*, vol. 133, pp. 15 – 29, 2015.

[42] G. Zhang, J. Jia, T-T Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 6, pp. 974–988, June 2009.

[43] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, April 2004.