UNIVERSITY OF CALIFORNIA
Los Angeles

Variational PDE-based Image Segmentation and Inpainting with Applications in Computer Graphics

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Kang-Yu Ni

2008

The dissertation of Kang-Yu Ni is approved.

Song-Chun Zhu

Luminita A. Vese

Andrea L. Bertozzi

Tony F. Chan, Committee Chair

University of California, Los Angeles

2008

To my mother

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGMENTS

Foremost, thanks go to my Ph. D. thesis advisor, Dr. Tony Chan, for his wise guidance and support throughout my time as a graduate researcher. I am very grateful for his encouragement and financial support, which gave me several opportunities to travel to conferences. Also appreciated were his efforts to make time for his students out of his busy schedule.

My sincere gratitude goes to my doctoral committee members, Dr. Andrea Bertozzi, Dr. Luminita Vese, and Dr. Song-Chun Zhu for their advice, time, and effort. I wish to express my admiration to Dr. Bertozzi and Dr. Vese, which motivates me to achieve my academic goals. I also appreciate their encouragement and help in my graduate studies. Special thanks go to Dr. Zhu for his willingness to be my summer internship mentor in 2007.

I owe a huge debt of gratitude to Dr. Selim Esedoglu for getting me started with my first Ph.D. research topic (histogram segmentation), especially his patient assistance on numerical implementations. He is like a second advisor, whose knowledge and guidance are helpful and deeply appreciated. I would also like to thank him for the opportunity to visit University of Michigan, Ann Arbor, to finish the segmentation project, during which time he spent numerous hours working with me.

I would like to express thanks to Dr. Doug Roble for his patient help during my summer internship at Digital Domain and for making the experience enjoyable. I had many inspiring discussions with him and gained a valuable perspective that is different from the UCLA applied math group. I would also like to thank him for his time and patience working with me on the presentation slides and rehearsing my talk several times. I couldn't have given a decent talk at SIGGRAPH without his help.

I have been very fortunate and grateful to have the opportunity to work with Dr.

Xavier Bresson, Dr. Byung-Woo Hong, and Dr. Sheshadri Thiruvenkadam on different research projects and appreciate their kindness, encouragements, and guidance. In addition, I appreciate Dr. Thiruvenkadam's help on writing my first research paper and also on revising my research statement for job applications. Dr. Hong has been very patient with me and continues to push our project forward.

I would like to thank Dr. Jerome Darbon, Dr. Doug Enright, Dr. Sung Ha Kang, and Dr. Berta Sandberg for their helpful discussions, suggestions, and encouragement.

Far too many colleagues to mention individually have assisted in so many ways at UCLA. They all have my sincere gratitude. In particular, I would like to thank Jason Chung, Julia Dobrosotskaya, Danielle O'Donnol, Frances Hammock, I-Shen Lai, Yanina Landa, Sunmi Lee, Tin Man Lee, Linh Lieu, Tungyou Lin, Ronald Lui, Nishali Metha, Frederick Park, Ya-Ju Tsai, Daisy Wang, Igor Yanovsky, and Mingqiang Zhu. I appreciate their generosity in many aspects of my graduate study and life. I would also like to thank all my instructors, in particular Dr. Inwon Kim, my TAs, the staff of IPAM and the Math Department, especially Maggie Albert, Rocie Carrillo, and Babette Dalton.

My greatest thankfulness goes to my family, especially my parents and sister for their continuous support. I thank my mother for her unrelenting love for our family. She is always there for me even though she is 6,780 miles away. My father always gives me valuable academic advice whenever I ask. My sister often gives me another perspective whenever things are not going well.

My final, and most heartfelt, acknowledgment must go to my incredibly supportive fiancé. I appreciate his generous love and patience with my countless writing questions. It would have been very difficult to make it through my graduate study without him.

This work was supported by ONR grant N00014-06-1-0345 and NSF grant DMS-

0610079.

VITA

1981	Born, Taiwan.
1999-2003	B.S. (Mathematics) with a minor in (Physics), University of California, Santa Barbara, California.
2003-2008	Ph.D. (Mathematics), University of California, Los Angeles, California.

PUBLICATIONS

Tony Chan, Selim Esedoglu and Kangyu Ni, *Histogram based Segmentation using Wasserstein Distances*, Proceedings of Scale Space and Variational Methods in Computer Vision, pp. 697-708, 2007.

Kangyu Ni, Doug Roble and Tony Chan, *A Texture Synthesis Approach to Elastica Inpainting*, ACM SIGGRAPH 2007 sketches.

Kangyu Ni, Sheshadri Thiruvenkadam and Tony Chan, *Matting through Variational Inpainting*, Proceedings of Signal and Image Processing, 2007.

Byung-Woo Hong, Stefano Soatto, Kangyu Ni, and Tony Chan, *Scale of Texture and its Application to Segmentation*, to appear in CVPR 08.

Kangyu Ni, Xavier Bresson, Selim Esedoglu, and Tony Chan, *Local Histogram based Segmentation using the Wasserstein Distance*, in preparation.

Kangyu Ni, Sheshadri Thiruvenkadam and Tony Chan, *Matting through Texture and Geometric Inpainting*, in preparation.

Kangyu Ni, Byung-Woo Hong, Stefano Soatto, and Tony Chan, *Unsupervised Multiphase Segmentation and Hierarchical Representation of Image Structure*, submitted.

Abstract of the Dissertation

Variational PDE-based Image Segmentation and Inpainting with Applications in Computer Graphics

by

Kang-Yu Ni

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2008

Professor Tony F. Chan, Chair

This dissertation explores the applications of variational PDE models to image processing, computer vision, and computer graphics, and also to building efficient numerical algorithms and schemes. In particular, the areas of contributions are segmentation, inpainting, and matting. There are three separate topics in segmentation. The first is unsupervised segmentation, in which we propose and analyze a nonparametric region-based active contour model for segmenting cluttered scenes. The novelty is the use of the Wasserstein distance in segmentation, which is able to measure the dissimilarity between histograms, either continuous or discontinuous, in a reasonable manner. We employ a fast global minimization method to solve the proposed model. An advantage of this method is that initializations can be arbitrary to obtain a global minimizer. Moreover, our proposed model has several properties due to the use of the Wasserstein distance. A variant of the proposed model is presented to handle local illumination changes in an image.

The second topic of segmentation examines the issue of scale in modeling texture for the purpose of segmentation. We propose a scale descriptor for texture and an energy minimization model to find the intrinsic scale of a texture at each location. For

each pixel, we use the intensity distribution of its local patch to determine the smallest size of the domain that can be used to generate neighboring patches. The obtained scale descriptor is applied for improving the segmentation model described above.

In the third topic of segmentation, we propose a multiphase segmentation algorithm based on Chan and Vese's two-phase piecewise constant segmentation model. The proposed algorithm recursively splits a partitioned region into two, starting from the largest scale, and automatically detects when all the regions cannot be partitioned further. The number of phases is not given prior and can be arbitrary, and the junctions of phase boundaries are implicitly dealt with. Additionally, the proposed model provides a full hierarchical representation of the structure of a image.

In the area of inpainting, we present a new technique that works well in both textured and non-textured areas of an image. Euler's elastica inpainting is a PDE-based variational model that works well for repairing smooth areas of an image while maintaining edge detail. However, it is slow due to a stiff, fourth order PDE and is difficult to control. On the other hand, texture synthesis techniques work well in inpainting for areas that contain repeating patterns. We combine these two techniques to accelerate and constrain the PDE solution. Instead of a stiff minimization, we have a combinatorial optimization problem that is quicker to solve.

In the area of matting, we propose a new algorithm that takes into account both texture and geometric structures of the foreground and background of the given image. We propose to utilize our inpainting algorithm for the matting problem, which extrapolates both geometric features and texture into unknown regions. The proposed matting algorithm improves previous algorithms, whose performance is uncertain in the presence of sharp discontinuities in the foreground and/or background.

CHAPTER 1

Introduction

The areas of contributions in this dissertation are segmentation, inpainting, and matting. Each chapter is based on a separate topic from these areas, although they are related. Chapter 2 is based on *Histogram based Segmentation using Wasserstein Distances* and *Local Histogram based Segmentation using the Wasserstein Distance.* An unsupervised segmentation model is proposed to separate two cluttered regions of a gray-scale image. Chapter 3 is based on *Scale of Texture and its Application to Segmentation.* A texture scale model is proposed to find at each location the intrinsic scale for texton, which is the smallest basic element of texture. The obtained scale is then applied to the histogram based segmentation shown in Chapter 2. Chapter 4 is based on *Unsupervised Multiphase Segmentation and Hierarchical Representation of Image Structure.* An unsupervised multiphase segmentation model is proposed that provides hierarchical representation of the structure of an image. Chapter 5 is based on *A Texture Synthesis Approach to Elastica Inpainting* and *Matting through Texture and Geometric Inpainting.* We develop an efficient numerical scheme for variational inpainting and also combine this algorithm with a texture synthesis technique so that it works for both texture and geometric areas of an image. Chapter 6 is based on *Matting through Variational Inpainting* and *Matting through Texture and Geometric Inpainting.* We take a new approach that utilizes variational inpainting within the matting problem.

In Chapter 2, we propose and analyze a nonparametric region-based active contour model for segmenting cluttered scenes. This model is unsupervised and assumes that pixel intensity is independently identically distributed. The proposed segmentation energy consists of a geometric regularization term that penalizes the length of region boundaries, and a region-based image term that uses the probability density function (or histogram) of pixel intensity to distinguish different regions. More specifically, the region data encourages partitioning the image domain so that the local histograms within each region are approximately uniform. The solutions of the proposed model do not need to differentiate histograms. The similarity between normalized histograms is measured by the *Wasserstein distance with exponent 1*, which is able to fairly compare two histograms, both continuous and discontinuous. We employ a fast global minimization method based on [4, 5] to solve the proposed model. The advantages of this method include less computational time compared with the standard minimization method by gradient descent of the associated Euler-Lagrange equation and the ability to find a global minimizer. Moreover, our proposed model has several desired properties due to the use of the Wasserstein distance. We further propose a variant of the model that addresses local illumination changes in an image.

Chapter 3 examines the issue of scale in modeling texture for the purpose of segmentation. We propose a scale descriptor for texture and an energy minimization model to find the scale of a given texture at each location. For each pixel, we use the intensity distribution in a local patch around that pixel to determine the smallest size of the domain that can be used to generate neighboring patches. The energy functional we propose to minimize is comprised of three terms: The first is the dissimilarity measure using the Wasserstein distance or Kullback-Leibler divergence between neighboring patch distributions; the second maximizes the entropy of the local patch, and the third penalizes larger size at equal fidelity. Our experiments show the proposed scale model successfully captures the intrinsic scale of texture at each location. We also apply our scale descriptor for improving texture segmentation based on the model in Chapter 2.

In Chapter 4, we propose an unsupervised multiphase segmentation algorithm based on Bresson et al.'s fast global minimization of Chan and Vese's two-phase piecewise constant segmentation model. The proposed algorithm recursively splits a partitioned region into two, starting from the largest scale, and automatically terminates and detects when all the regions cannot be partitioned further. The number of phases is not given and can be arbitrary, and the junctions of phase boundaries are implicitly dealt with. Additionally, the proposed model provides a full hierarchical representation of the structure of a given image. Experimental results show that the recursive segmentation successfully partitions the given images according to region scale and contrast in an intuitive way.

In Chapter 5, we present a new technique for wire and scratch removal (inpainting) that works well in both textured and non-textured areas of an image and also efficiently extend our algorithm for multiple frames for either static or moving camera/objects. [6] introduced a technique for inpainting using an Euler's elastica energy-based variational model that works well for repairing smooth areas of the image while maintaining edge detail. The technique is very slow due to a stiff, 4th order PDE and difficult to control. [7] used texture synthesis techniques for inpainting and hole filling. This works well for areas of an image that contain repeating patterns. We have combined these two techniques to accelerate and constrain the solution of the fourth order PDE. Instead of a stiff minimization, we have a combinatorial optimization problem that is much quicker to solve and gets to similar solutions of elastica inpainting. Furthermore, we combine this algorithm with texture synthesis by a threshold to fully make use of both repeating patterns and geometry in both space and time.

In Chapter 6, we propose a new matting algorithm that takes into account both texture and geometric structures of the foreground and background of the given image. The matting problem extracts the object of interest of an image with an accurate transparency (or matte) of the object. This is an under-constrained problem because there are too many unknowns. Previous matting algorithms impose priors on the unknowns and work well for many natural images. However, their performance is uncertain in the presence of sharp discontinuities in the foreground and/or background regions. This is because those algorithms are based on statistical estimates of the unknowns and geometry is not considered. In response to this shortcoming, we propose to employ our proposed inpainting algorithm in Chapter 5 to solve the matting algorithm. Our experimental results show that the proposed matting algorithm accurately extracts the matte of an image with geometric and/or texture structures.

CHAPTER 2

Local Histogram based Segmentation using the Wasserstein Distance

2.1 Introduction

Image segmentation plays an important role in computer vision. The process involves partitioning the image domain into several regions either according to edge information or region information so that the image within each region has uniform characteristics. The characterized regions depend on the application and may include one or more of the following: edges, intensities, textures, and shapes. Snake [8], balloon [9], and geodesic active contours based [10, 11] methods use edge detection functions and evolve contours towards sharp gradients of pixel intensity. This classic active contour approach is widely used in medical imaging. However, it is not robust to noise because noise also has large gradients. Typically a noisy image has to be smoothed, which may lose important edge information. Region-based active contours incorporate region and boundary information and are robust to noise. Furthermore, they are able to detect objects with either sharp or smooth edges. One of the first region-based active contours is the Mumford-Shah segmentation model [12], which approximates an image by a piecewise smooth function, with a length penalizing term. However, this model is difficult to solve in practice because of the edge set. The active contours without edges (ACWE) model [1], a variant of the piecewise constant Mumford-Shah model, approximates an image by a two-phase piecewise constant function and is based on a level-set implementation [13]. The minimizing flow is derived by computing the variation of the energy with respect to the level set function. Region competition [14] is a statistical and variational model that is based on minimizing a generalized Bayes and Minimum description length criterion. The model penalizes the boundary length and the Bayes error within each region, in which appropriate probability distributions are chosen. The ACWE, region competition, and other parametric region-based active contour models, such as [15, 16], assume the probability density function (pdf) of the pixel intensity in each region up to a few parameters. For example, often a Gaussian distribution is assumed with mean and variance the only unknowns. However, many natural images are not necessarily described by a specific distribution. Nonparametric region-based active contour models, such as [2, 3, 17, 18], use the full pdf, or histogram, of the intensity to drive the segmentation. Therefore, they do not suffer from the above limitations. Our model is related to, yet different from, previous work. In [18, 17], the segmentation model is supervised and the data descriptors directly depend on the regions, which consequently involves histogram differentiation in the evolution equations. Unsupervised segmentation models in [2, 3] take an information-theoretic approach and their data descriptors also directly depend the regions. In our work, each pixel is initially assigned with a local histogram, i.e. a normalized histogram of the pixel intensities in a neighborhood of that pixel. The model finds a partition such that the local histograms in each region are similar to one another. We use an optimal transport distance to measure the similarity between histograms.

Previous models are quite effective in segmenting images when the histograms in each region are distinct. However, the distances used for comparing histograms are pointwise and may not be reliable even under simple circumstances. Furthermore, some distances used are not metrics; for instance, triangle inequality is not satisfied. As an example of this issue, the pointwise distance between two delta functions with disjoint supports is the same no matter how far apart the supports are; this is a situation that arises often in segmentation applications, since for example images consisting of two objects with approximately constant but different intensities would fall into this category. Previous nonparametric approaches did not address this issue and used the Parzen window method [19] to approximate and smooth histograms. The degree of smoothness has to be controlled by a user-selected parameter. To overcome this issue, we propose to use an optimal transport distance to compare histograms, which extends as a metric to measures such as the delta function. We believe this to be the more natural and appropriate way to compare histograms.

The optimal transport, or the Monge-Kantorovich problem, is to find the most efficient plan to rearrange one probability measures into another. We will introduce Kantorovich's version [20] here. Let (X, μ) and (Y, ν) be two probability measure spaces. Let π be a probability measure on the product space $X \times Y$ and $\Pi(\mu, \nu) = \{\pi \in P(X \times Y) : \pi[A \times Y] = \mu[A]$, and $\pi[X \times B] = \nu[B]$ hold for all measureable sets $A \in X$ and $B \in Y\}$ be the set of admissible transference plans. For a given cost function $c : X \times Y \to \mathbb{R}$, the total transport cost, associated to plan $\pi \in \Pi(\mu, \nu)$, is

$$I[\pi] = \int_{X \times Y} c(x, y) d\pi(x, y).$$

The optimal transport cost between μ and ν is

$$T_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} I[\pi].$$

More details can be found in [21] and [22]. In the case of probability measures μ and ν on $\mathbb{R}$, with cost function $c(x, y) = |x - y|^p$, the optimal transport cost has a closed form,

$$T_p(\mu, \nu) = \int_0^1 |F^{-1}(t) - G^{-1}(t)|^p dt,$$

where F and G are the cumulative distribution functions of μ and ν, respectively and F^{-1} and G^{-1} represent their corresponding inverse functions. The optimal transport distance, commonly called the *Wasserstein distance with exponent p*, is $W_p(\mu, \nu) = T_p(\mu, \nu)^{1/p}$. When the cost function is Euclidean distance $c(x, y) = |x - y|$,

$$W_1(\mu, \nu) = \int_0^1 |F^{-1}(t) - G^{-1}(t)| dt = \int_{\mathbb{R}} |F(x) - G(x)| dx.$$

The last equality is obtained by Fubini's Theorem. The Wasserstein distance defines a metric and is insensitive to oscillations [21].

The main contributions of this chapter are as follows:

1. the novelty of using the Wasserstein distance to properly compare histograms, both discontinuous and continuous,

2. a segmentation model that does not need to differentiate histograms to find a solution,

3. the use of the fast global minimization method [5] to solve the proposed model, which significantly improves the previous model [23] in two ways, the computational time is less than the standard method and initialization can be arbitrary,

4. mathematical properties of the proposed model are presented.

The goal of this work is to understand low-order feature segmentation, which is based on the statistics of image intensity and does not consider high-order features, such as gradient, curvature, orientation and scale. We specifically discuss a nonparametric region-based active contour variational model in [23] and derive its mathematical properties. We also discuss its limitations.

2.2 Related Works

Kim et al. took an information-theoretic approach and proposed a nonparametric region-based active contours model. Given an image $I : \Omega \to [0, L]$ with two regions, in each of which pixel intensities are independently identically distributed, a curve $\overrightarrow{C}$ is evolved towards the boundary. The region inside (resp. outside) the curve $\overrightarrow{C}$ is denoted by R_+ (resp. R_-). Define the region labels associated with curve $\overrightarrow{C}$ by

$$L_{\overrightarrow{C}}(x) = \begin{cases} L_+ & \text{if } x \in R_+ \\ L_- & \text{if } x \in R_-. \end{cases}$$

The proposed model maximizes the mutual information between the image pixel intensities and region labels, subject to a constraint on the total length of the region boundaries:

$$\inf_{\overrightarrow{C}} \qquad \lambda \oint_{\overrightarrow{C}} ds - |\Omega| M(I(X); L_{\overrightarrow{C}}(X)), \tag{2.1}$$

where λ is a positive parameter and

$$M(I(X); L_{\overrightarrow{C}}(X)) = h(I(X)) - h(I(X)|L_{\overrightarrow{C}}(X)).$$

Since the entropy of the image $h(I(X))$ is constant, maximizing the mutual information between $I(X)$ and $L_{\overrightarrow{C}}(X)$ minimizes the conditional entropy $h(I(X)|L_{\overrightarrow{C}}(X))$. The curve $\overrightarrow{C}$ is evolved so that knowing which region a pixel belongs to decreases the uncertainty of the pixel intensity. The conditional entropy is

$$\begin{aligned}
&h(I(X)|L_{\overrightarrow{C}}(X)) \\
&= -\frac{1}{|\Omega|}\Big(\int_{R_+}\log P_+(I(x))dx + \int_{R_-}\log P_-(I(x))dx\Big),
\end{aligned}$$

where the probability density functions $P_+(I(x))$ and $P_-(I(x))$ of each region are approximated using the Parzen window method [19],

$$P_+(I(x)) = \frac{1}{|R_+|}\int_{R_+} K(I(x) - I(\hat{x}))d\hat{x}, \tag{2.2}$$

$$P_-(I(x)) = \frac{1}{|R_-|}\int_{R_+} K(I(x) - I(\hat{x}))d\hat{x}. \tag{2.3}$$

The Gaussian function $K(z) = (1/\sqrt{2\pi\sigma^2})e^{-z^2/2\sigma^2}$ is used as a smoothing kernel, where σ is a scalar parameter that controls the smoothness of the approximation. The minimization problem (2.1) is solved by the following gradient flow:

$$\begin{aligned}
\frac{\partial \overrightarrow{C}}{\partial t} \quad = \quad & \Big[\log\frac{P_+(I(\overrightarrow{C}))}{P_-(I(\overrightarrow{C}))} + \frac{1}{|R_+|}\int_{R_+}\frac{K(I(x) - I(\overrightarrow{C}))}{P_+(I(x))}dx \\
& -\frac{1}{|R_-|}\int_{R_-}\frac{K(I(x) - I(\overrightarrow{C}))}{P_-(I(x))}dx\Big]\overrightarrow{N} - \lambda\kappa\overrightarrow{N},
\end{aligned} \tag{2.4}$$

where $\overrightarrow{N}$ is the outward normal and κ is the curvature of $\overrightarrow{C}$. The level-set method with narrow band approach was used for implementation.

Herbulot et al. also took a nonparametric region-based active contours approach and used information entropy as competition between two regions:

$$\inf_{\overrightarrow{C}} \qquad \lambda\oint_{\overrightarrow{C}} ds + h(I(X), R_+) + h(I(X), R_-), \tag{2.5}$$

where the entropy of pixel intensities in each region is

$$h(I(X), R_+) = -\int_{R_+} P_+(I(x)) \log P_+(I(x)) dx$$
$$h(I(X), R_-) = -\int_{R_-} P_-(I(x)) \log P_-(I(x)) dx.$$

The probability density functions $P_+(I(x))$ and $P_-(I(x))$ are approximated using the Parzen window method as described in (2.2) and (2.3). The minimization is solved by the following gradient flow:

$$\begin{aligned}
\frac{\partial \overrightarrow{C}}{\partial t} &= \Bigg[-\Big(P_+(\log P_+ + 1) - P_-(\log P_- + 1) \Big) \\
&- \frac{1}{|\Omega|} \Bigg(h(I(X), R_+) - h(I(X), R_-) \\
&\quad + \int_{R_+} K(I(x) - I(\overrightarrow{C})) \log P_+(I(x)) dx \\
&\quad + \int_{R_-} K(I(x) - I(\overrightarrow{C})) \log P_-(I(x)) dx \Bigg) \Bigg] \overrightarrow{N} - \lambda \kappa \overrightarrow{N},
\end{aligned}$$

The curve evolution is implemented by using smoothing B-splines.

2.3 Proposed Model I

In this section, we discuss an unsupervised segmentation model proposed in our previous work [23] for cluttered images. Suppose the observed gray-scale image $I : \Omega \to [0, L]$ is measurable and has two regions of interests and the pixel intensity in each region is independently identically distributed. Denote by $\mathcal{R}_{x,r}$ the ball of radius r centered at x. For a Lebesgue-measurable subset S of $\mathbb{R}^2$, denote $|S|$ its 2-dimensional

Lebesgue measure, i.e. its area. Define the local histogram of a pixel $x \in \Omega$ by

$$P_x(y) := \frac{|\{z \in \mathcal{R}_{x,r} \cap \Omega : I(z) = y\}|}{|\mathcal{R}_{x,r} \cap \Omega|},$$

for $0 \leq y \leq L$. Define the corresponding cumulative distribution function by

$$F_x(y) := \frac{|\{z \in \mathcal{R}_{x,r} \cap \Omega : I(z) \leq y\}|}{|\mathcal{R}_{x,r} \cap \Omega|}, \tag{2.6}$$

for $0 \leq y \leq L$. These are the image data used in the following proposed segmentation model:

$$\inf_{\Sigma, P_1, P_2} \left\{ E_1(\cdot, \cdot, \cdot | I) = \text{Per}(\Sigma) + \lambda \int_{\Sigma} W_1(P_1, P_x) dx + \lambda \int_{\Sigma^c} W_1(P_2, P_x) dx \right\}, \tag{2.7}$$

where Per(Σ) is the perimeter of the set Σ. This minimization problem finds an optimal region $\Sigma \subseteq \Omega$ and approximates the local histograms inside Σ (resp. Σ^c) by a constant histogram P_1 (resp. P_2). Recall that W_1 is the Wasserstein distance with exponent 1, described in the introduction:

$$W_1(P_1, P_2) = \int_0^L |F_1(y) - F_2(y)| dy. \tag{2.8}$$

Energy functional (2.7) can be formulated in terms of the level set method [13]. The boundary between Σ and Σ^c is represented by the 0-level set of a Lipschitz function

$\phi : \Omega \to \mathbb{R}$.

$$\inf_{\phi, F_1, F_2} \quad \Big\{ E_1(\cdot,\cdot,\cdot|I) = \int_\Omega |\nabla H(\phi(x))| dx \tag{2.9}$$
$$+\lambda \int_\Omega H(\phi(x)) \int_0^L |F_1(y) - F_x(y)| dy\, dx$$
$$+\lambda \int_\Omega [1 - H(\phi(x))] \int_0^L |F_2(y) - F_x(y)| dy\, dx \Big\},$$

where H is the Heaviside function, $\int_\Omega |\nabla H(\phi(x))| dx$ represents Per(Σ), and $H(\phi)$ (resp. $1 - H(\phi)$) defines Σ (resp. Σ^c).

The minimization of (2.9) can be achieved by a standard two-step scheme. First, we fix ϕ and minimize with respect to F_1 and F_2, respectively. Variations with respect to F_1 and F_2 yield the following optimality conditions that should be held for all $0 \le y \le L$,

$$\int H(\phi(x)) \frac{F_1(y) - F_x(y)}{|F_1(y) - F_x(y)|} dx = 0$$

and

$$\int [1 - H(\phi(x))] \frac{F_2(y) - F_x(y)}{|F_2(y) - F_x(y)|} dx = 0,$$

respectively. Therefore,

$$F_1(y) = \text{median of } F_x(y), \text{ over } \{x : \phi(x) \ge 0\} \tag{2.10}$$

and

$$F_2(y) = \text{median of } F_x(y), \text{ over } \{x : \phi(x) < 0\}. \tag{2.11}$$

Next, with fixed $F1$ and $F2$, the gradient descent of Euler-Lagrange equation for ϕ

gives

$$\phi_t = \delta(\phi)\left[\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \lambda \int_0^L (|F_1(y) - F_x(y)| - |F_2(y) - F_x(y)|)dy\right], \quad (2.12)$$

where δ is a regularized Dirac function and $\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right)$ is the curvature of the level sets. Steps (2.10), (2.11), and (2.12) are iterated alternately, until convergence to a steady state solution.

However, numerically, equation (2.12) has serious time-step restrictions. The curvature term can be approximated by

$$\frac{\partial}{\partial x}\left(\frac{\phi_x}{\sqrt{\phi_x^2 + \phi_y^2 + \epsilon^2}}\right) + \frac{\partial}{\partial y}\left(\frac{\phi_y}{\sqrt{\phi_x^2 + \phi_y^2 + \epsilon^2}}\right),$$

where $\epsilon > 0$ so that the denominators are not zero but small enough to stay close to the solution. Therefore, the time-step restriction of the explicit scheme for (2.12) is [24] $\triangle t \leq c \cdot \epsilon \cdot (\triangle x)^2$, where c is a constant. This time-step restriction can be improved to $\triangle t \leq c \cdot (\triangle x)^2$ with Chambolle's method [25], which is presented in Section 2.4.3.

2.4 Fast Global Minimization of Model I

2.4.1 Global Minimization of Model I

Like many variational segmentation models, model (2.7) suffers from being non-convex (with respect to Σ) and is therefore sensitive to initializations. The requirement of reasonable initializations conflicts the purpose of automatic segmentation. Numerically, a non-compactly supported dirac function is used in [1] to increase the chances of finding global minimizers of the piecewise constant segmentation model. Theoretically, based on the framework of [5, 4, 26], we propose the following global minimization of Model I:

$$\min_{0\leq u\leq 1, P_1, P_2}\Big\{E_2(\cdot,\cdot,\cdot|I) = \int_\Omega |\nabla u(x)|dx \quad + \quad \lambda\int_\Omega W_1(P_1,P_x)u(x)dx \qquad (2.13)$$
$$+ \quad \lambda\int_\Omega W_1(P_2,P_x)(1-u(x))dx\Big\}.$$

This problem is equivalent to problem (2.7) but overcomes the non-convexity. Let 1_S denote the characteristic function of set S. This model (2.13) extends the original minimization over the non-convex set $\{u \in BV(\Omega) : u = \mathbf{1}_\Sigma$ for some set Σ with finite perimeter$\}$ to the convex set $\{u \in BV(\Omega) : 0 \leq u \leq 1\}$. Thus, (2.13) is convex with respect to u and, unlike (2.7), does not have (non-global) local minima with respect to the geometric unknown.

The major advantage of (2.13) is that initializations can be arbitrary. The relation between (2.7) and (2.13) is that, for fixed F_1 and F_2, a global minimizer of (2.7) can be found through a global minimizer of (2.13). This relation is stated in the following theorem, which is based on the geometric properties of TV.

Theorem 1: (Global Minimizers) Suppose $I(x) \in [0,1]$. If P_1, and P_2 are fixed, and $u(x)$ is any minimizer of $E_2(\cdot, P_1, P_2|I)$, then for a.e. $\mu \in [0,1]$, $\mathbf{1}_{\{x:u(x)>\mu\}}(x)$ is a global minimizer of $E_1(\cdot, P_1, P_2|I)$.

Proof: By the coarea formula and setting $\Sigma(\mu) := \{x : u(x) > \mu\}$, we can write E_2 in terms of E_1 in the following [4],

$$
\begin{aligned}
E_2(u, P_1, P_2|I) &= \int_0^1 \left\{ \text{Per}(\Sigma(\mu)) + \lambda \int_{\Sigma(\mu)} W_1(P_1, P_x)dx \right. \\
&+ \left. \lambda \int_{\Omega-\Sigma(\mu)} W_1(P_2, P_x)dx \right\} d\mu \\
&= \int_0^1 E_1(\Sigma(\mu), P_1, P_2|I)d\mu,
\end{aligned}
$$

Therefore, if u is a minimizer of $E_2(\cdot, P_1, P_2|I)$, then for a.e. $\mu \in [0,1]$, $\Sigma(\mu)$ is a minimizer of $E_1(\cdot, P_1, P_2|I)$.$\square$

Variations of E_2 with respect to F_1 and F_2 yield the following optimality conditions that should hold for all $0 \leq y \leq L$:

$$\int u(x) \frac{F_1(y) - F_x(y)}{|F_1(y) - F_x(y)|} dx = 0$$

and

$$\int [1 - u(x)] \frac{F_2(y) - F_x(y)}{|F_2(y) - F_x(y)|} dx = 0,$$

respectively. Therefore,

$$F_1(y) = \text{weighted (by } u(x)\text{) median of } F_x(y), \tag{2.14}$$

and

$$F_2(y) = \text{weighted (by } 1 - u(x)\text{) median of } F_x(y), \tag{2.15}$$

Minimizing E_2 with respect to u is postponed until Sec. 2.4.3.

2.4.2 Existence of Global Minimization Solutions

In this section, we show the existence of a minimizer for and convexity of model (2.13).

Theorem 2: (Existence of Solutions) For fixed P_1 and P_2,

$$\begin{aligned}\min_{0\leq u\leq 1}\Big\{E_2(\cdot,P_1,P_2 \quad | \quad I) = &\int_\Omega |\nabla u(x)|dx \\ &+\lambda\int_\Omega W_1(P_1,P_x)u(x)dx \\ &+\lambda\int_\Omega W_1(P_2,P_x)(1-u(x))dx\Big\}\end{aligned} \tag{2.16}$$

has a solution $u \in BV(\Omega)$ with $0 \leq u \leq 1$.

Proof: Let $\{u_n\} \in BV(\Omega)$ with $0 \leq u \leq 1$ be a minimizing sequence. Then, $\int_\Omega |Du_n|$ is uniformly bounded. Since every uniformly bounded sequence in $BV(\Omega)$ is relatively compact in $L^1(\Omega)$, there exists a subsequence $\{u_{n_k}\}$ converging to some $u \in BV(\Omega)$. Since $u_{n_k} \to u$ in $L^1(\Omega)$, we have $u_{n_k} \to u$ in measure, i.e. $|\{x : |u_{n_k}(x) - u(x)| \geq \epsilon\}| \to 0$ as $\epsilon \to 0$. Since we also have $0 \leq u_{n_k} \leq 1$, u satisfies $0 \leq u \leq 1$. Finally, we can check that u is indeed a minimizer. By the lower semicontinuity of $BV(\Omega)$,

$$\int_\Omega |Du| \leq \liminf_{k\to\infty} \int_\Omega |Du_{n_k}|.$$

By Fatou's lemma,

$$\int_\Omega W_1(P_1,P_x)u(x)dx \leq \liminf_{k\to\infty}\int_\Omega W_1(P_1,P_x)u_{n_k}(x)dx$$

and

$$\int_\Omega W_1(P_2, P_x)(1 - u(x))dx \leq \liminf_{k\to\infty} \int_\Omega W_1(P_2, P_x)(1 - u_{n_k}(x))dx.$$

Therefore,

$$E_2(u, F_1, F_2|I) \leq \liminf_{k\to\infty} E_2(u_{n_k}, F_1, F_2|I).$$

$\square$

We will next show that $E_2[u, P_1, P_2|I]$ is convex with respect to each variable. First, E_2 is convex with respect to u because $\int_\Omega |Du(x)|dx$ is convex in u and the set $\{u \in BV(\Omega) : 0 \leq u \leq 1\}$ is convex. Moreover,

Theorem 3: The minimization problem

$$\min_{P_1 \in P(\Omega)} E_2[u, \cdot, P_2|I]$$

is convex, where $P(\Omega)$ denotes the set of Borel probability measures on Ω.

Proof: $E_2[u, \cdot, P_2|I]$ is convex in P_1 because the Wasserstein distance is a metric and in particular satisfies the triangle inequality. Since $P(\Omega)$ is a convex set, minimization with fixed u and P_2 is a convex problem.$\square$

Similarly, the minimization $\min_{P_2 \in P(\Omega)} E_2[u, P_1, \cdot|I]$ is convex.

2.4.3 Fast Minimization

Minimizing the proposed energy E_2 in (2.13) with respect to u can be efficiently solved by applying methods in [27, 5]. The regularization and data terms in (2.13) can be decoupled by using a new variable v to replace u in the data term and adding a convex term that forces v and u to be the same:

$$\min_{u,0\leq v\leq 1}\int_\Omega|\nabla u(x)|dx+\frac{1}{2\theta}\int_\Omega(u(x)-v(x))^2dx \\ +\lambda\int_\Omega r(x,F_1,F_2)v(x)dx\ , \tag{2.17}$$

where

$$r(x,F_1,F_2)=\int_0^L|F_1(y)-F_x(y)|-|F_2(y)-F_x(y)|dy,$$

and $\theta > 0$ is a small parameter. Minimizing the convex variational model (2.17) can be approached by alternately solving the following coupled problems:

$$\min_u\int_\Omega|\nabla u(x)|+\frac{1}{2\theta}(u(x)-v(x))^2dx \tag{2.18}$$

$$\min_{0\leq v\leq 1}\int_\Omega\frac{1}{2\theta}(u(x)-v(x))^2+\lambda r(x,F_1,F_2)v(x)dx \tag{2.19}$$

Minimization in (2.18) can be achieved fast by Chambolle's method [25], based on the dual formulation of the total variation norm:

$$u(x)=v(x)-\theta\text{div}p(x)\ , \tag{2.20}$$

where $p=(p^1,p^2)$ solves $\nabla(\theta\text{div}p-v)-|\nabla(\theta\text{div}p-v)|p=0$ and is solved by a fixed point method,

$$p^{n+1}=\frac{p^n+\delta t\nabla(\text{div}p^n-v/\theta)}{1+\delta t|(\text{div}p^n-v/\theta)|}. \tag{2.21}$$

The solution of (2.19) is found as in [5]:

$$v(x) = \max\{\min\{u(x) - \theta\lambda r(x, F_1, F_2), 1\}, 0\} \ . \tag{2.22}$$

Our fast minimization scheme is to iterate (2.14), (2.15), (2.21), (2.20), and (2.22) alternately, until convergence.

2.5 Proposed Model II

We propose a variant of Model I that handles segmentation properly when the captured image has uneven lighting exposure, due to reasons such as the location of the light source and camera. The original model considers the data term globally, i.e. compares all the local histograms within each region. Therefore, when the local lighting changes significantly, local histograms of the same feature may have similar shapes but are far apart by a translation in the intensity axis. As a result, the Wasserstein distance between them is large and thus the original model is not designed to deal with uneven lighting. To model this variation, we introduce a function $a(x)$, representing the translation in the intensity axis, and propose a new model:

$$\begin{aligned}\inf_{\Sigma,a,F_1,F_2} \Big\{ E_3(\Sigma, a, F_1, F_2|I) = \mathrm{Per}(\Sigma) + & \frac{\alpha}{2}\int |\nabla a(x)|^2 dx \\ & +\lambda \int_\Sigma \int_0^L |F_1(y) - F_x(y - a(x))| dy\ dx \\ & +\lambda \int_{\Sigma^c} \int_0^L |F_2(y) - F_x(y - a(x))| dy\ dx \Big\}.\end{aligned} \tag{2.23}$$

This model allows local histograms to translate on the intensity axis in order to find a best fit among one another within each region. A regularity constraint $\int |\nabla a(x)|^2 dx$

is imposed to ensure smoothness of a.

To solve this minimization problem, we have the following three-step scheme. The evolution equations for F_1, F_2 and ϕ can be derived similarly as before:

$$F_1(y) = \text{median of } F_x(y - a(x)), \text{ over } \{\phi \geq 0\} \tag{2.24}$$

$$F_2(y) = \text{median of } F_x(y - a(x)), \text{ over } \{\phi < 0\} \tag{2.25}$$

$$\begin{aligned} \phi_t = \delta(\phi)\Bigg[\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda\int_0^L \Big(&|F_1(y) - F_x(y - a(x))| \\ &-|F_2(y) - F_x(y - a(x))|\Big)dy\Bigg] . \end{aligned} \tag{2.26}$$

The minimization with respect to $a(x)$ is to solve:

$$\begin{aligned} \inf_a E_3(\Sigma, \cdot, F_1, F_2|I) = \frac{\alpha}{2}\int |\nabla a(x)|^2 dx \\ +\lambda\int_\Sigma\int_0^L |F_1(y) - F_x(y - a(x))|dy\, dx \\ +\lambda\int_{\Sigma^c}\int_0^L |F_2(y) - F_x(y - a(x))|dy\, dx . \end{aligned} \tag{2.27}$$

Without the first term, $a(x)$ can be solved explicitly by

$$a_0(x) = \begin{cases} F_1^{-1}(0.5) - F_x^{-1}(0.5) & \text{if } \phi(x) > 0 \\ F_2^{-1}(0.5) - F_x^{-1}(0.5) & \text{if } \phi(x) \leq 0 \end{cases}$$

Therefore, the problem of (2.27) is transformed into solving the following:

$$\inf_{a} \frac{1}{2} \int |a(x) - a_0(x)|^2 dx + \frac{\alpha}{2} \int |\nabla a(x)|^2 dx \ . \tag{2.28}$$

The solution to (2.28) is $a(x) - \alpha \triangle a(x) = a_0(x)$, which can be solved by fast fourier transform. We may also employ the fast global minimization technique for Model II, instead using (2.26).

2.6 Properties of Proposed Models

Our model has several desired mathematical properties as shown in Table 1. In Sec. 2.4.2, we show the existence of solution and convexity of model in each variable. In the discrete sense, if the resolution of an image f is $m \times n$ and L is the number of gray levels, then $u \in \mathbb{R}^{m \times n}$ and $P \in P(\{0, 1, ...L\}) \subset \mathbb{R}^L$. Therefore, the model in the discretized form is convex in $\mathbb{R}^{m \times n} \times P(\{0, 1, ...L\})$ and thus a global minimizer can be found. Based on Chambolle's dual method regarding the length-penalizing term, the solution converges after a small number of iterations, compared to directly solving the associated Euler-Lagrange equation. Moreover, since the Wasserstein distance is insensitive to oscillations, our model is intrinsically robust to noise. On the other hand, it does not require histograms to be smoothed, which has to be done for many segmentation models even for noiseless images. For instance, the Wasserstein distance is able to distinguish the distance between any pair of delta functions with disjoint supports. Many distances do not tell apart the distance between two disjointly supported histograms unless the histograms are smoothed. The complexity of computing one iteration is O(Lmn). For a 200×150 image, the computational time for a solution to converge is approximately two minutes. Since the partition is implicitly embedded in function u, the model is able to handle topological changes.

Kim et al.'s model also has existence of solution. Their model minimizes over a non-convex set $\{u \in BV(\Omega) : u = 1_\Sigma$ for some set Σ with finite perimeter$\}$, thus does not guarantee to get a global minimizer. The gradient flow (2.4) has a curvature term and the convergence can be slow, due to the CFL condition discussed in Sec. 2.3. However, the fast global minimization method may be applicable to their model. The probability density functions are estimated by the Parzen window method. This enables their model to handle noise but introduces a user-selected parameter, i.e. kernel width. They use the fast Gauss transform to compute probability densities, which reduces the complexity of computing one iteration to O(M), where M is the size of the narrow band. The level-set method is used for curve evolution and thus allows topological changes.

Herbulot et al. use smoothing B-splines to implement their derived evolution equation instead of the usual level-set method to avoid extensive computational time. The complexity of each iteration is O(LM), where L is the number of gray levels and M and the size of the narrow band. The parametric method using B-splines does not handle topological changes of the contours. They further use a smoothing B-splines in order to be more robust to noise. The tradeoff between the smoothness and interpolation error is controlled by a parameter that has to be chosen by the user. Their model also minimizes over a non-convex set, thus does not guarantee to get a global minimizer.

2.7 Experimental Results

2.7.1 Comparison with other methods

Our model does not require histograms to be smoothed for proper performance. In contrast, Parzen window method [19] requires a parameter selection to estimate prob-

Table 2.1: Properties of the proposed, Kim et al. [2], and Herbulot et al. [3] models

	Our model	Kim et al. [2]	Herbulot et al. [3]
existence of solution	✓	✓	✓
global minimum/convexity	✓	x	x
fast minimization	✓	x	✓
insensibility to noise	✓	-	-
no need to smooth histograms	✓	x	x
local change of lighting	✓	x	x
complexity for one iteration	O(Lmn)	O(M)	O(LM)
computational time	2 mins	10 mins	10 mins
handle topological changes	✓	✓	x

ability density functions. For instance, if the width of the kernel is too small, pointwise metrics cannot detect similar intensities. In Fig.2.1, we have a synthetic image with three regions, in each of which the pixel intensity is independently identically distributed. The middle and outer regions are similarly distributed and their corresponding images look similar. A desired partition is to group these two regions together and distinguish them from the inner region. In (b), we see that the final contour of our model captures this. On the other hand, without smoothing histograms, the final contour (a) of the model in [2] falsely groups the inner and middle regions together. This is because the histograms of the inner and middle regions overlap 50% but the histograms of the middle and outer region do not overlap.

We emphasize here how nonparametric models are able to deal with a greater variety of images than parametric models. In this experiment, the object and background have the same intensity mean and variance. In Fig.2.2(a), we show the boundaries of the objects in red curves and the corresponding histograms in each region. Fig.2.2(c) and (b) are the final contours of our proposed model and ACWE, respectively. The proposed model is able to distinguish the objects from the background, as well as other nonparametric models, such as [2, 3, 17, 18] (not shown). On the other hand the

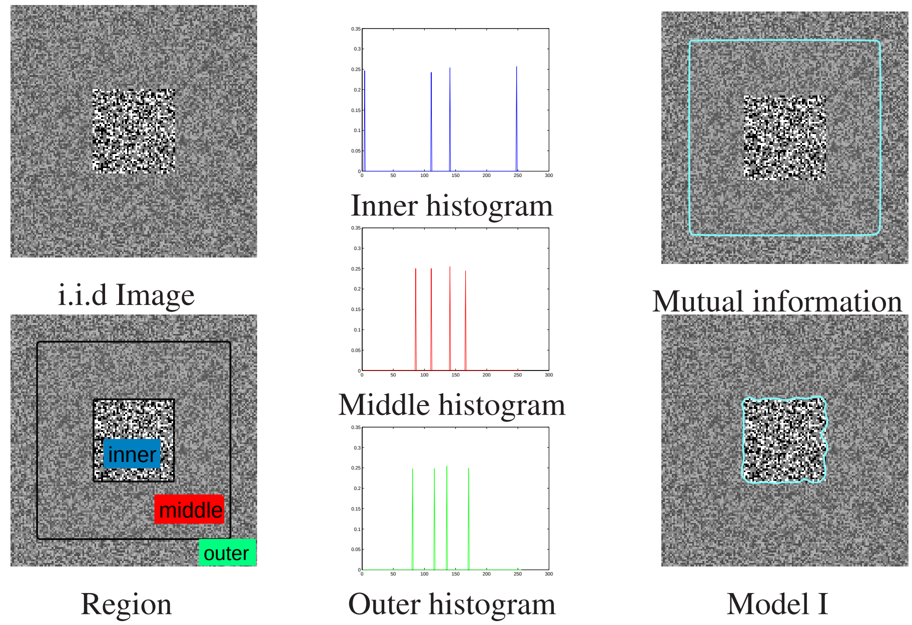
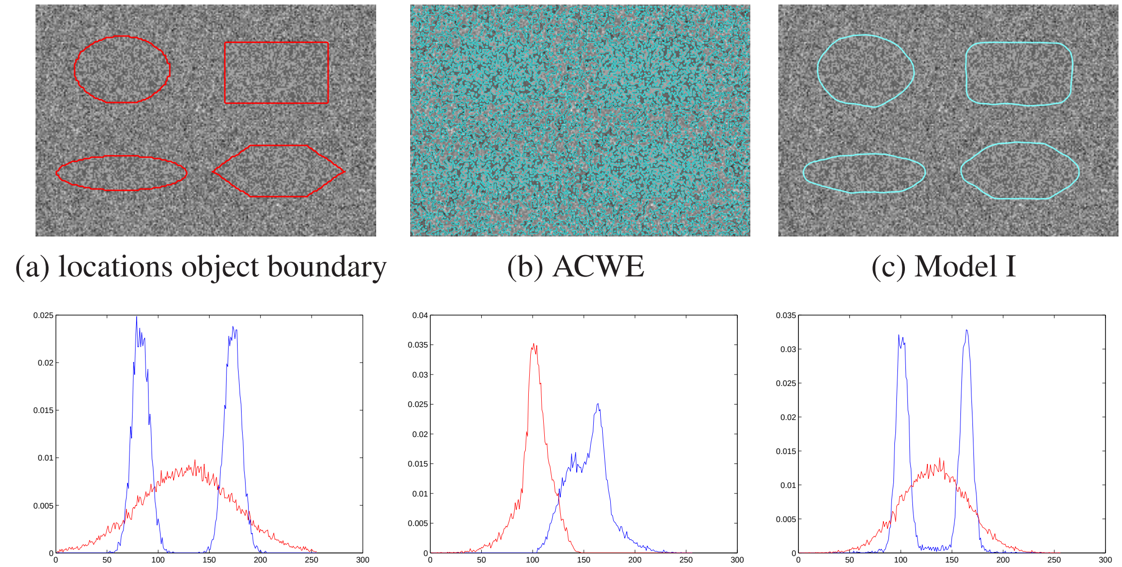


Figure 2.1: The given image, shown in upper left corner, has three regions, in which pixel intensity is independently identically distributed. The inner and outer regions look similar, as well as their corresponding histograms and cumulative distributions. Wasserstein distance does not require histograms to be smoothed in order to properly compare histograms. Upper right shows the final contour of Kim et al.'s model when the histograms are not smoothed. Bottom right shows the final contour of proposed model I.

ACWE model cannot handle this case due to its parametric nature.

2.7.2 Comparison between original model and fast global minimization

The proposed fast global minimization improves the original minimization in [23] of model I. Fig.2.3 is a downsized 175×135 image of cheetah. In Sec. 2.4.1, we explain that the global minimization model does not have local minima and thus is guaranteed to find a global minimum. We experiment with several images with different and arbitrary initializations and all arrive at similar results. This is a nice consequence of the global minimization model being convex with respect to each variable. On the other hand, the original minimization is non-convex and thus requires initializations



(a) locations object boundary (b) ACWE (c) Model I

(d) histograms on each region (e) ACWE (f) Model I

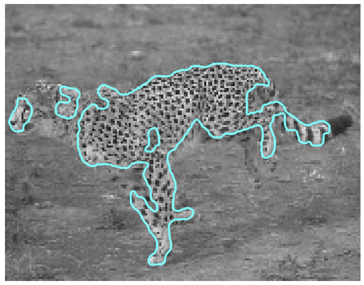
Figure 2.2: Objects and background regions have the same intensity mean and variance. (b) Final contours of ACWE model. (c) Final contours of proposed Model I.

to be reasonably close to the final contours. Moreover, the fast global minimization improves the speed from two hours to two minutes.

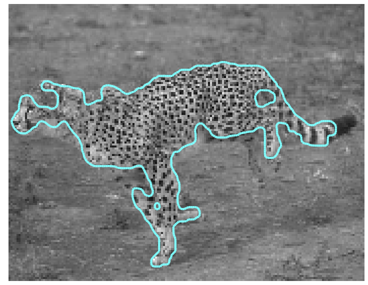
2.7.3 Robustness to noise/More results of Model I and Model II

Fig.2.4(a) is a clean image of cheetah and (b) is with noise. The final contour shown in (d) by the global minimization of Model I is able to segment the cheetah patterns and is nearly as good as the result in (c) of the clean image. Since noisy images need more data for local histograms, the neighborhood size needs to be large enough. In this experiment, the neighborhood radius is 11.

Fig.2.5 shows other experimental results of Model I. The first experiment is a 285×281 image consisting of two Brodatz textures. The final contours are shown in (a) and the corresponding histograms on each region are plotted in (c). Model I is able distinguish these two Brodatz textures, even though their intensity distributions are



(a) result of [23]



(b) result of (2.13)

Figure 2.3: Down-sized cheetah image. Global minimization improves segmentation result.

highly discontinuous. The second is a 481×321 image of tiger; (b) shows the final contours by Model I and (d) shows the histograms in each region. The final contour successfully selects the tiger patterns.

Fig.2.6 shows that Model II improves Model I when there are local lighting changes in the image. The first experiment is on a 384×223 image of cheetah. In (a), we see that Model I is able to capture some of the cheetah patterns but not near the back legs, due to the local lighting difference. Final contours of Model II, in (b), are more accurate. Another experiment is a 282×218 image of fish. The final contours by Model I, in (d), do not select the fish patterns accurately, because the local illumination is significantly uneven. Model II, on the other hand, is able to overcome this difficulty, as shown in (e) the final contours separates the fish patterns from the background.

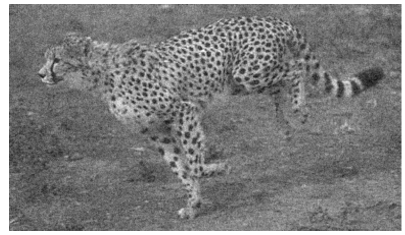
2.7.4 Implementation issues

We show a method to solve the weighted median for $F_1(y)$ in equation (2.14) in the discrete case.

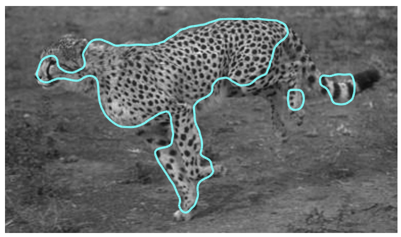
For each $y = 0, 1, ..., L$,

1. Compute the weighted histogram H_y of value $F_x(y)$, with weight $u(x)$. More

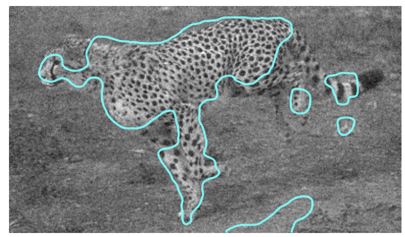
(a) clean cheetah image



(b) noisy cheetah image

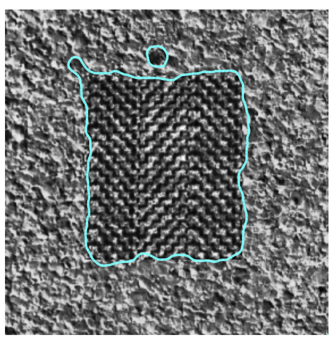


(c) final contour of clean image

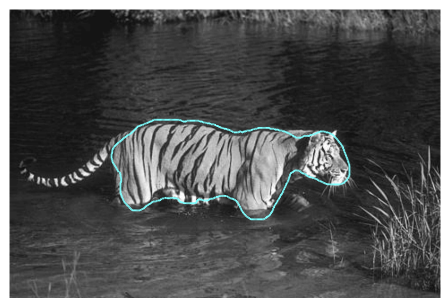


(d) final contour of noisy image

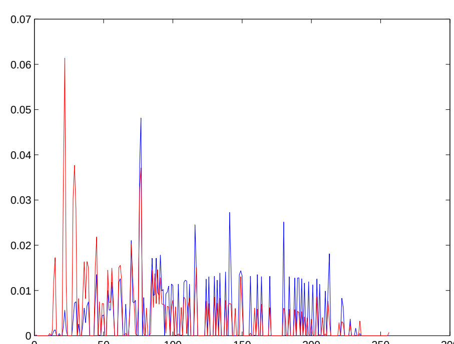
Figure 2.4: The performance of Model I is nearly as good even with added noise to the original image.



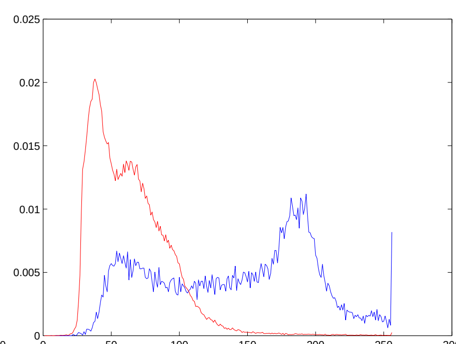
(a) Model I



(b) Model I

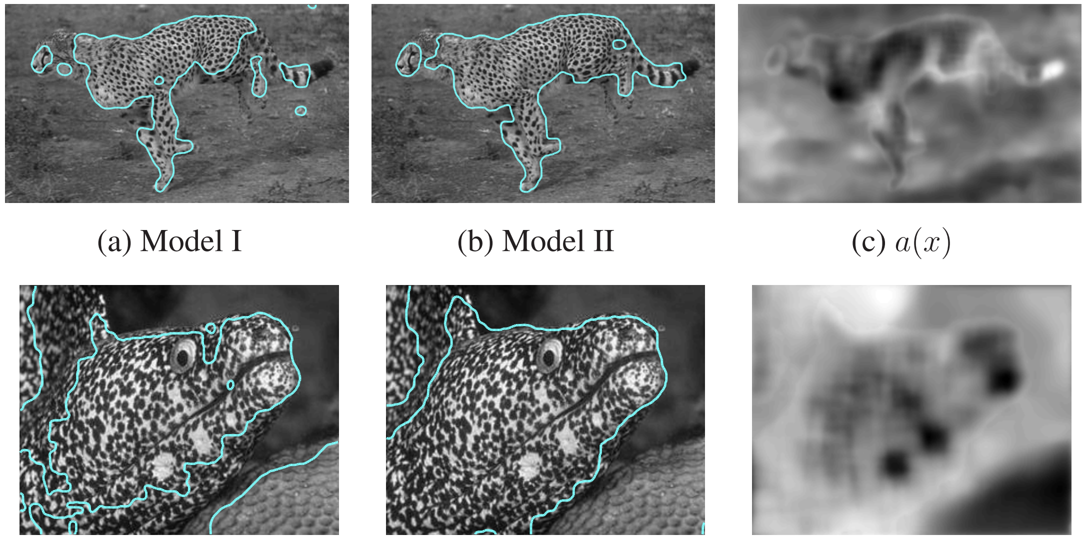


(c) histograms



(d) histograms

Figure 2.5: Final contours of Model I.



(a) Model I (b) Model II (c) $a(x)$

(d) Model I (e) Model II (f) $a(x)$

Figure 2.6: The smoothness component of Model II (b) allows local illumination changes and captures more of the cheetah patterns than Model I (a). Similarly, Model I decently segments the fish patterns (d), and Model II is able to improve the result (e).

precisely, for all pixels $x \in \Omega$, each $F_x(y)$ is counted $u(x)$ times. Then, the weighted histogram H_y is normalized by dividing by the total count $\sum_{x\in\Omega} u(x)$.

2. For each weighted histogram H_y, compute the cumulative distribution C_y.

3. The weighted median is then $F_1(y) = C_y^{-1}(0.5)$.

The calculation of $F_2(y)$ is similar and with weight $1 - u(x)$.

We empirically demonstrate that segmentation results are not sensitive to the size of local neighborhood histograms, within a reasonable range. The experiment is on a 384×223 image of cheetah, shown in Fig.2.4(a). Fig.2.7 shows final contours by global minimization of Model I with different neighborhood sizes, radius ranging from 1 to 25. If the neighborhood size is smaller than the clutter features, the final contour partitions clutter features into smaller regions, an undesired result. If the neighborhood size is large enough, our results show the cheetah patterns are segmented for a large range of neighborhood sizes.

2.7.5 Limitations and extensions

Our segmentation model is formulated for gray-scale images but can be extended to color images. The data term can be generalized because the Wasserstein distance is defined on any space of probability measures. However, the implementation would be much more complicated because there is no closed form for the Wasserstein distance between probability measures on Euclidean spaces with dimensions larger than one. The Earth Movers Distance between signatures is equivalent to the Wasserstein distance when signatures have the same total mass (or normalized discrete pdfs) and the optimization has been investigated in [28]. This can be a possible direction to extend our segmentation model. Works in [29, 30] numerically solve the the optimal maps of the optimal transport problem on $\mathbb{R}^2$ and may also be applied to our extension. Another limitation is that our model assumes the given image has two regions of clutters. Many natural images have more than two regions and requires a multi-phase segmentation model. This limitation can be overcome, since our model has a natural extension to multi-phase segmentation as in [31]. Moreover, since our model has a minimal assumption on the intensity probability density, it does not take into account higher-order characteristics, such as gradient, scale, and orientation. For example, if two textures have the same intensity probability density, our model is not able to distinguish them. Therefore, suitable characteristics have to be added in order distinguish different regions. In addition, our segmentation model can be contributed to existing segmentation algorithms [32] that consider many image characteristics, including clutters.

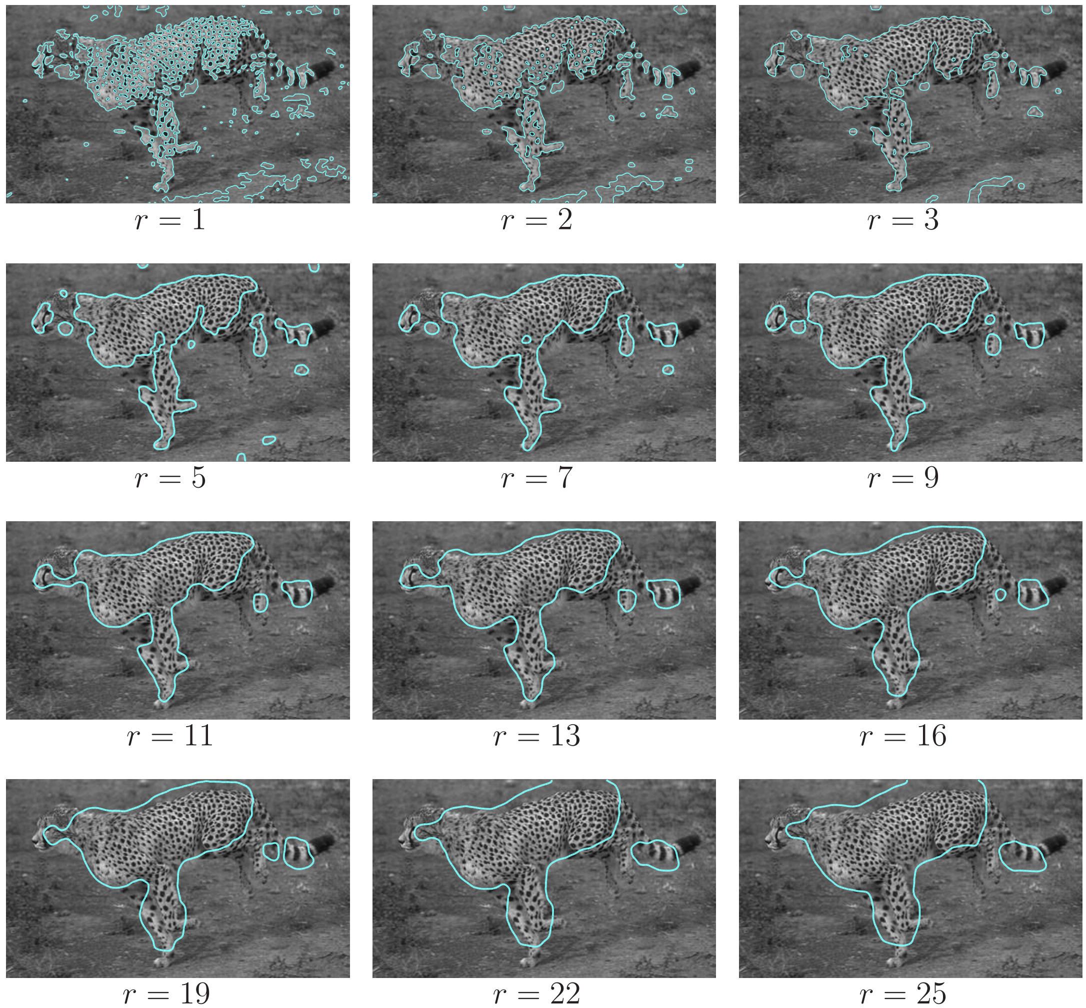


Figure 2.7: The neighborhood size in model (2.13) needs to be equal or bigger than the smallest features of interest in the given image. The segmentation results are not too sensitive to the size r of the neighborhood, but are more accurate when the size is closer to that of the smallest image features of interest.

2.8 Conclusion

In this chapter, we proposed a fast global minimization of a local histogram based model using the Wasserstein distance with exponent 1 to segment cluttered images. Our model is different from previous nonparametric region-based active contours in two ways. The first is the use of the Wasserstein distance, which is able to compare both continuous and discontinuous histograms properly. The second is that the proposed model does not need to differentiate histograms to find the solutions. We have proved a number of desired mathematical properties of the model and provided experimental verifications. In the future, we will generalize our model to color images and multi-phase segmentation. The former can be achieved by using the fast minimization of vectorial total variation in [33] and adapting the numerical scheme for computing the optimal transport distance in [28, 29, 30]. The later can be approached by applying methods such as the multi-phase level set framework [31].

CHAPTER 3

Scale of Texture and its Application to Segmentation

3.1 Introduction

A "texture" is a region of the image that exhibits stationary – or cyclostationary – statistics of some sort. If one were to compute the histogram in a region around each pixel, there would be some function of this histogram that is either constant (in practice slowly-varying) or periodic as we move the pixel within the texture. Because the local statistics are pooled from a region around each pixel, a fundamental question in the definition, design, or classification of texture is the area of this region, or "scale". Some statistics are only stationary when computed at a certain scale, but not at larger and/or smaller scales. The "right" scale thus defines the texture and plays an important role, recognized early in the pioneering work of Julesz [34, 35], with many subsequent attempts to define "elementary texture elements".

Textures are important in the analysis of images, as they provide a mid-level representation that is robust to the actual realization (pixel values) [36, 37, 38, 39], so that "segments" of the image that have a consistent texture can be used as "tokens" [40, 41, 42]; this is also important in image modeling, compression and synthesis [43, 7, 44]. An arsenal of different analytical tools has been brought to bear in the analysis of textures, from statistical models to filtering methods, to geometric approaches. Zhu et al. [43] model texture as a Markov random field (MRF), or equivalently the Gibbs distribution. Efros and Leung [7] observe that textures range in between regular (repeating) and stochastic (without explicit textels) and many synthesis methods often fail in preserving the geometric structures. Their synthesis method is based on a statistical non-parametric model that preserves spatial locality. Inspired by Julesz, Zhu et al. and Wu et al. [45, 46] take a mathematical approach and identify a texture by an equivalence class of statistical features. They later connect this idea with MRF texture models by a minimax entropy scheme [47].

In this work, we address the issue of scale in textures head-on. As Zhu et al. [48] point out, the basic texture element, also referred to as "texton" in the MRF literature and considered a fundamental token for pre-attentive visual perception [35], remains a vague concept in need of a better formalization. We provide a characterization of scale that is not restricted to simple statistics, but instead – in a generative framework – we see it as the generator, or "seed," of a texture using any generative model. Rather than texture modeling and classification, therefore, we focus our attention entirely on determining the size a texton in a given image.

The scale descriptor in this work corresponds to the texton size or texture scale. Many previous works define scale in relation to certain diffusion operators or filters. Lindeberg [49] associates scale with the size of intensity gradient and uses the Gaussian kernel to examine the local scale at each pixel. Brox and Weickert [50] and Strong et al. [51] define scale based on the region size a pixel belongs to. They observe that under the total variation regularization, the intensity change in a pixel is inversely proportional to the region size. In [50], scale is defined as the time taken for a feature to disappear under the TV flow and is applied to accomplish difficult texture segmentation. In [51], scale is the inverse of the intensity change under the TV denoising model [52]. SIFT [53] describes local features in an image by taking the differences of blurred-images that are obtained by convolving with Gaussian filters with different variances. These definitions of scale do not take into account the neighborhood statistics so that they cannot provide an intrinsic texture scale that measure the smallest repetitive pattern locally. For regular (or repeated) textures, scale is the size of the smallest image patch that generates a texture by repeating the patch side by side. Wolf et al. [42] use a patch matching criterion to find texture edges and then incorporate it into a region based active contour model for texture segmentation. Their texture map is successful for segmentation but does not reveal any signs of the correct texton size. For stochastic textures, the spatial relation may not be found and thus may not be obtained by simply stitching textons together. Instead, we take a non-parametric approach and use the entire distribution of the patch to find a texton's size.

For stationary textures, the intrinsic scale is the size of the smallest domain where the distribution is close to that of any other domain of the same size within the texture. Because in practice the statistics may not be strictly stationary, but slowly-varying instead, in practice we look for the smallest local patch whose probability density function (pdf) is similar to the one computed on its neighboring local patches (which we later call "neighboring patch" for short).

We introduce an intrinsic scale in modeling of texture and use it to improve segmentation models. The intrinsic scale is not uniform across the image domain. This is in contrast to many schemes for texture segmentation where local pdfs are compared, for instance using the Wasserstein distance [54], but they are computed on a local domain the size of which is fixed throughout the image. If the selected size is smaller than the texton, these schemes over segment the texture; if it is too large, the segmentation may not be accurate because local patches cross over texture boundaries. Not only is the texton size not constant across regions, it may even vary within a texture region, albeit slowly. We believe that by automatically finding the intrinsic scale, histogram-based segmentation will improve its performance. Additionally, the scale can also be added into the data term to distinguish two textons with the same pdf but different scales. Huang et al. [55] also use scale as feature for segmentation. They use local patch's pdf to find a best natural scale of textons. However, our model is significantly different in two ways. The first is that our scale finds the intrinsic scale of texture which is obtained by the size of texton whereas their scale gives a local feature for segmentation but is not necessary the size of texton that is a basic element of texture. The second is that our segmentation model is a convex minimization problem in a variational framework, in which initialization can be arbitrary, whereas they use a probabilistic model that requires proper initializations along with feature given by filter response.

3.2 Texture Scale

3.2.1 Notations

Let $I : \Omega \subset R^2 \to [0,1]$ be an observed gray-scale image. Define the local patch $\mathcal{R}_{x,r}$ around the pixel point $x = (x_1, x_2) \in \Omega$ with size r ("radius" in analogy to circles) by:

$$\mathcal{R}_{x,r} = \{z \in \Omega \mid \max_{1,2}\{|x_1 - z_1|, |x_2 - z_2|\} < r\} \tag{3.1}$$

Define the neighboring patch of the local patch by:

$$\mathcal{N}_{x,r} \quad = \quad \mathcal{R}_{x,3r} \backslash \mathcal{R}_{x,r}. \tag{3.2}$$

The local histogram, $h_{\mathcal{R}}(y)$, on $\mathcal{R}$ counts the number of pixels whose intensity is $y \in [0,1]$:

$$h_{\mathcal{R}}(y) = \int_{\mathcal{R}} \delta(y - I(x))dx, \tag{3.3}$$

where δ is Dirac's Delta.

The probability density function (or normalized histogram), $P_{\mathcal{R}}$, on $\mathcal{R}$ is the probability of a pixel having value $y \in [0, 1]$:

$$P_{\mathcal{R}}(y) = \frac{\int_{\mathcal{R}} \delta(y - I(x))dx}{\int_{\mathcal{R}} dx}. \tag{3.4}$$

In this chapter, histograms are normalized. The cumulative distribution function, $F_{\mathcal{R}}$, describes the probability of a pixel having value less than y, for all $y \in [0, 1]$:

$$F_{\mathcal{R}}(y) = \int_0^y P_{\mathcal{R}}(t)dt. \tag{3.5}$$

The Wasserstein distance with exponent 1 between two probability density functions P_1 and P_2 is:

$$D_W(P_1, P_2) = \int_0^1 |F_1(y) - F_2(y)|dy, \tag{3.6}$$

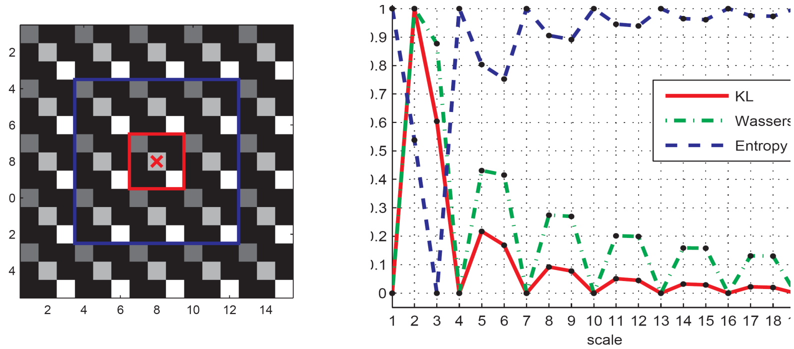
where F_1 and F_2 are the corresponding cumulative distribution functions.

The Kullback-Leibler divergence D_{KL} from P_1 to P_2 is:

$$D_{KL}(P_1||P_2) = \int_0^1 P_1(y) \log \frac{P_1(y)}{P_2(y)} dy. \tag{3.7}$$

The entropy of P is:

$$H(P) = -\int_0^1 P(y) \log P(y) dy. \tag{3.8}$$



(a) Texture (b) Energy plot

Figure 3.1: Synthetic texture example: (a) Local patch at 'X' (inside red) and neighboring patch (between red and blue). (b) Energy vs. patch size. Red: histogram difference using the Kullback-Leibler divergence. Green: histogram difference using the Wasserstein distance. Blue: entropy of the local patch histogram

3.2.2 Description of the scale model

Our proposed scale descriptor is derived by energy minimization of the following model:

$$\inf_{r} D(P_{\mathcal{R}_{x,r}}, P_{\mathcal{N}_{x,r}}) - \alpha H(P_{\mathcal{R}_{x,r}}) + \beta r(x), \tag{3.9}$$

where α and β are positive design parameters. In the first term, D is an appropriate measure of the dissimilarity between two probability distributions; for example, we use both the Wasserstein distance D_W and the Kullback-Leibler divergence D_{KL} in this chapter. The first term of this energy functional measures the difference between the pdf on the local patch and the pdf on the neighboring patch. Minimizing the difference finds a size whose local patch satisfies the histogram matching criterion. The second term maximizes the entropy of $P_{\mathcal{R}_{x,r}}$, the complexity of the histogram on the local patch. This term avoids selecting homogeneous patches as textons despite their small difference in the pdf with their neighborhood. The third term penalizes the size r to find the smallest one among all the ones that satisfy the criterion. To understand the proposed model, we show a synthetic texture example and plot the first and second terms versus the patch size r, at the indicated pixels. Fig.3.1 (a) shows a local patch (in red) around pixel 'X' and a neighboring patch (in between the blue and red curves). In (b), we look at how the first and second terms of (3.9) change with respect to r. The green and red curves are the first terms with the Wasserstein distance and Kullback-Leibler divergence, respectively. The blue curve is the entropy of the histogram on the local patch, whose maxima (patch being most complex) appear periodically when r is a multiple of the texton size. Minima (satisfy histogram matching criterion) appear periodically at multiples of r. Therefore, the correct scale should be the smallest one among all arguments of the minimum. In this example, the texton size is 1, or a 3×3 patch. The entropy term is redundant in this example but is necessary in general when there are homogeneous areas within the texton.

Fig.3.2 is an example consisting of two synthetic textures, on each of which we select two pixels (A, B and C, D), one closer to the texture edge than the other. From the energy plots, we see that the entropy increases rapidly with patch size as soon as the patch begins to overlap both texture regions. Therefore, measuring the complexity of a local patch's histogram alone is not sufficient to find the scale. The distance between the histograms on the local patch and neighboring patch also increases rapidly as the local patch begins to overlap both textures, indicating the correct texton size has already been passed. This shows a appropriateness of using the histogram matching criterion.

The proposed model (3.9) finds the local scale of a texture. However, it may not be accurate at locations near texture edges, due to the nature of patches. Fig.3.3 (a) marks three locations, one at the left texture, one near the texture edge, and one on the right texture. The histogram differences by both Wasserstein distance in (c) and

Kullback-Leibler divergence in (d) attain local minima periodically because both local and neighboring patches are almost symmetric about the texture edge when the patch size is large. Therefore, histogram comparison must be modified in order to find the correct scale especially for the pixels in the vicinity of the boundary of different textures. We propose the following modification of model (3.9):

$$\inf_r D^*(P_{\mathcal{R}_{x,r}}, P_{\mathcal{N}_{x,r}}) - \alpha H(P_{\mathcal{R}_{x,r}}) + \beta r(x) \tag{3.10}$$

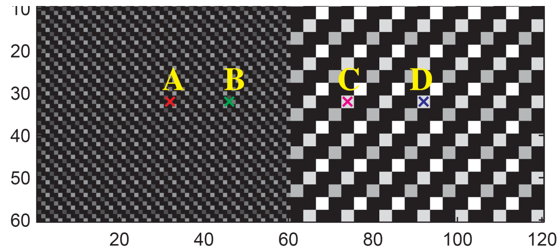
and

$$D^*(P_{\mathcal{R}_{x,r}}, P_{\mathcal{N}_{x,r}}) = \min_i D(P_{\mathcal{R}_{x,r}}, P_{\mathcal{N}_{x,r,i}}), \tag{3.11}$$

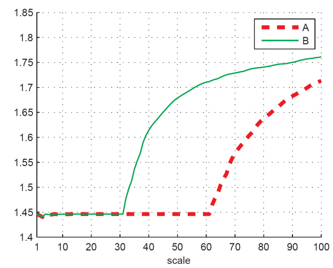
and $\mathcal{N}_{x,r,i}$ is a sub-neighboring patch within $\mathcal{N}_{x,r}$ whose size is r. For computational efficiency, 8 sub-neighboring patches are pre-defined as follows:

$$\{\mathcal{R}_{(x_1+2r,x_2+2r),r}, \mathcal{R}_{(x_1,x_2+2r),r}, \mathcal{R}_{(x_1-2r,x_2+2r),r}, \mathcal{R}_{(x_1-2r,x_2),r},$$
$$\mathcal{R}_{(x_1-2r,x_2-2r),r}, \mathcal{R}_{(x_1,x_2-2r),r}, \mathcal{R}_{(x_1+2r,x_2-2r),r}, \mathcal{R}_{(x_1+2r,x_2),r}\}$$

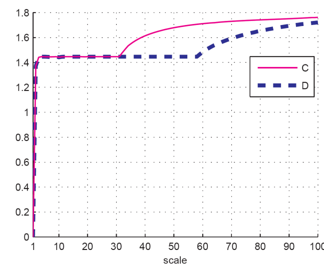
Numerically, the proposed models are solved in the discrete setting, instead of the standard PDE method that derives the Euler-Lagrange equations of the energy functionals (3.9) and (3.10), followed by the steepest descent. This is because r is a discrete variable. Moreover, as also seen in the energy plots, the proposed model has many local minima, thus the steepest descent method does not find a global minimum.



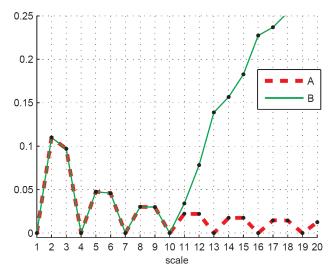
(a) Two textures



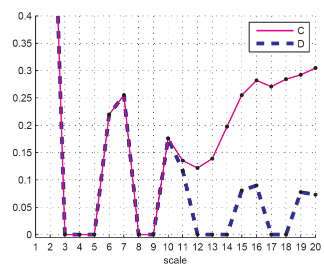
(b) Entropy



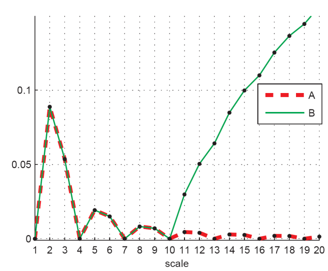
(c) Entropy



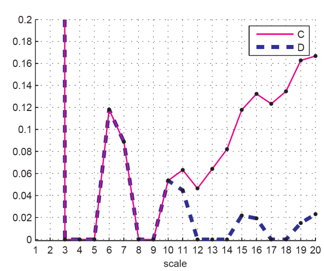
(d) Wasserstein



(e) Wasserstein

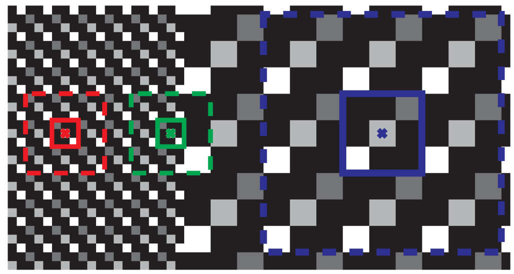


(f) KL divergence

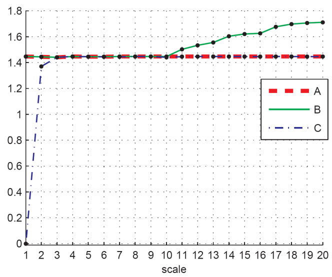


(g) KL divergence

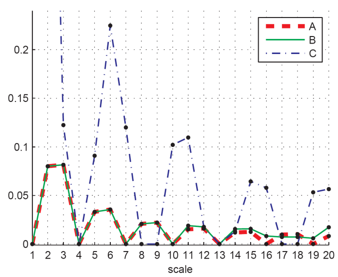
Figure 3.2: Image consisting of two synthetic textures. (a) Mark two locations A, B on the left texture and two locations C, D on the right. (b) Entropy vs. size of local patch at A and B. (c) Entropy vs. size of local patch at C and D. (d) Histogram difference vs. size with Kullback-Leibler divergence at A and B. (e) Histogram difference vs. size with Kullback-Leibler divergence at C and D. (f) Histogram difference vs. size with Wasserstein distance at A and B. (g) Histogram difference vs. size with Wasserstein distance at C and D



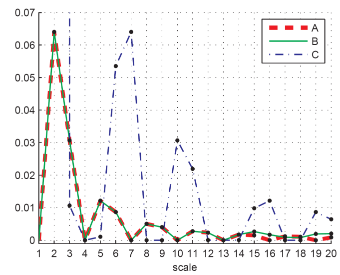
(a) Location B at texture edge



(b) Entropy

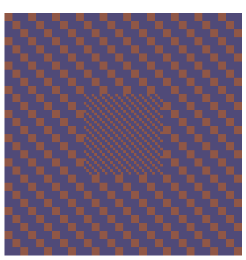


(c) Wasserstein

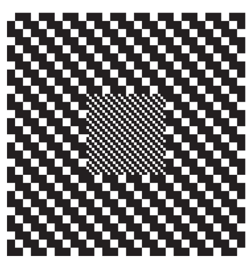


(d) KL divergence

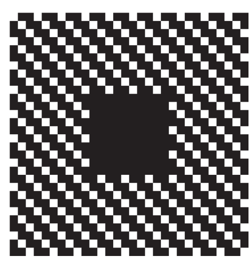
Figure 3.3: Image consisting of two synthetic textures. (a) Mark three locations A on the left texture, B near the texture edge and B on the right. (b) Entropy vs. size of local patch at each location. (c) Histogram difference vs. size with Kullback-Leibler divergence at each location. (d) Histogram difference vs. size with Kullback-Leibler divergence at each location.



(a) Texture



(b) Intensity



(c) Scale 1

(d) Scale 4

Figure 3.4: (a) Image consisting of two synthetic textures with the same histogram but different scales. (b) Segmentation by using intensity [1]. (c) Histogram based segmentation with scale $r = 1$. (d) Histogram based segmentation with scale $r = 4$.

3.3 Texture Segmentation

In this section, we utilize scale and propose an unsupervised texture segmentation model. Our model is adapted from the histogram based segmentation [54], a two-phase nonparametric region-based active contour that uses local histograms as image features. The model partitions the image domain into two regions so that the local histograms within each region are homogeneous. In [54], the local histograms have a uniform patch size and we propose to use an adaptive scale. In addition, we use scale as an image feature in the segmentation model. We give an example to show that scale plays an important role. Fig.3.4 (a) is an image consisting of two textures with the same histogram but different scales. The segmentation result in (b) is by the two-phase piecewise constant active contour model [1], indicated by the intensities black and white. The partition is within textons and does not distinguish textures, because two textures have the same intensity mean. In (c), we show the partition using histogram based segmentation model with global scale $r = 1$. The partition captures the inner texture but also falsely includes partial outer texture, because the scale is too small for the outer region. In (d), the global scale $r = 4$ is large enough and two textures are considered the same because they have the same histogram. To distinguish them, scale has to be added as an image feature in the segmentation model.

Our proposed model uses scale for characterizing histograms and also as an image feature, as shown in the following:

$$\begin{aligned}\min_{0\leq u\leq 1, P_1, P_2, r_1, r_2} \int_\Omega |\nabla u| \quad &+ \quad \int_\Omega [\lambda_1 D_W(P_1, P_{x,r(x)}) + \lambda_2 (r_1 - r(x))^2] u(x) dx \\ &+ \quad \int_\Omega [\lambda_1 D_W(P_2, P_{x,r(x)}) + \lambda_2 (r_2 - r(x))^2] (1 - u(x)) dx,\end{aligned} \tag{3.12}$$

where λ_1 and λ_2 are positive parameters. Minimizing this energy functional separates the image domain into two so that the local histograms within each region are homogeneous and the scale intensities are homogeneous within each region. The first term penalizes the total length of the object boundary. The second and third are fidelity terms. The partition can be obtained by the following thresholding: $\Omega = \{u \leq 0.5\} \cup \{u > 0.5\}$. P_1 and P_2 are the optimal histograms in each region; r_1 and r_2 are the approximated scale constants in each region.

The minimization of (3.12) can be approximated by a three-step scheme, using the methods in [54] and [56]. First, we fix u, r_1, and r_2 and minimize with respect to F_1 and F_2. The optimality conditions yield

$$\int u(x) \frac{F_1(y) - F_{x,r(x)}(y)}{|F_1(y) - F_{x,r(x)}(y)|} dx = 0$$

and

$$\int [1 - u(x)] \frac{F_2(y) - F_{x,r(x)}(y)}{|F_2(y) - F_{x,r(x)}(y)|} dx = 0,$$

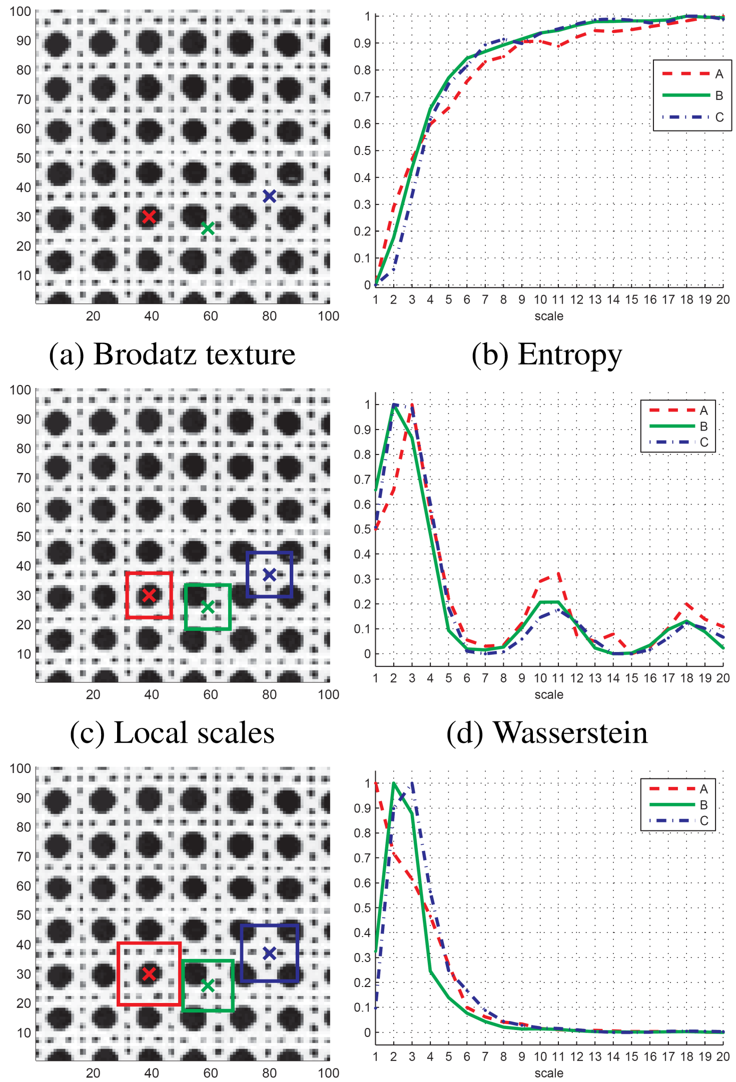
respectively, for each $0 \leq y \leq L$. Therefore,

$$F_1(y) = \text{weighted median of } F_{x,r(x)}(y) \text{ with weight } u(x), \tag{3.13}$$

and

$$F_2(y) = \text{weighted median of } F_{x,r(x)}(y) \text{ with weight } (1 - u(x)). \tag{3.14}$$

Second, fixing u, F_1, and F_2 and minimizing with respect to r_1 and r_2 gives



(a) Brodatz texture (b) Entropy

(c) Local scales (d) Wasserstein

(e) Local scales (f) KL divergence

Figure 3.5: Brodatz texture image. (a) Mark three locations A, B, and C. (b) Entropy vs. size of local patch at A, B, and C. (c) Selected scales by proposed model using Wasserstein distance at A, B, and C. (d) Histogram difference vs. size with Wasserstein distance at A, B, and C. (e) Selected scales by proposed model using Kullback-Leibler divergence at A, B, and C. (f) Histogram difference vs. size with Wasserstein distance at A, B, and C. (g) Histogram difference vs. size with Kullback-Leibler divergence at A, B, and C.

$$r_1 = \int_\Omega r(x)u(x)dx / \int_\Omega u(x)dx, \tag{3.15}$$

and

$$r_2 = \int_\Omega r(x)(1-u(x))dx / \int_\Omega (1-u(x))dx. \tag{3.16}$$

Third, fixing F_1 and F_2, minimization in u can be solved efficiently by using the methods in [25] and [56]. The regularization term and the data terms in (3.12) can be decoupled by adding a new variable v in a convex term:

$$\min_{u, 0 \leq v \leq 1} \int_\Omega |\nabla u(x)|dx + \frac{1}{2\theta}\int_\Omega (u(x)-v(x))^2 dx + \int_\Omega f(x)v(x)dx \ ,$$

where $f(x) = \lambda_1 \int_0^L |F_1(y) - F_{x,r(x)}(y)| - |F_2(y) - F_{x,r(x)}(y)|dy + \lambda_2[(r_1 - r(x))^2 - (r_2 - r(x))^2]$, and θ is a scalar parameter that is sufficiently small.

The convex minimization problem (3.17) can be solved the following coupled problems, alternately:

$$\min_u \int_\Omega |\nabla u(x)| + \frac{1}{2\theta}(u(x)-v(x))^2 dx \tag{3.17}$$

and

$$\min_{0 \leq v \leq 1} \frac{1}{2\theta}\int_\Omega (u(x)-v(x))^2 dx + \int_\Omega f(x)v(x)dx \ . \tag{3.18}$$

Equation (3.17) can be solved efficiently by the Chambolle's method [25], based on the dual formulation of the total variation norm, $\int_\Omega |\nabla u(x)| dx = \sup \left\{ \int_\Omega u \text{ div } p \; dx \;\middle|\; p \in C_c^1(\Omega; \mathbb{R}^2) : |p(x)| \leq 1, \forall x \in \Omega \right\}$. The solution is

$$u(x) = v(x) - \theta \text{ div} p(x) \ , \tag{3.19}$$

where p solves the equation $\nabla(\theta \text{ div} p - v) - |\nabla(\theta \text{ div} p - v)| p = 0$, which is solved by a fixed point method,

$$p^{n+1} = \frac{p^n + \delta t \nabla(\text{div} p^n - v/\theta)}{1 + \delta t |(\text{div} p^n - v/\theta)|}. \tag{3.20}$$

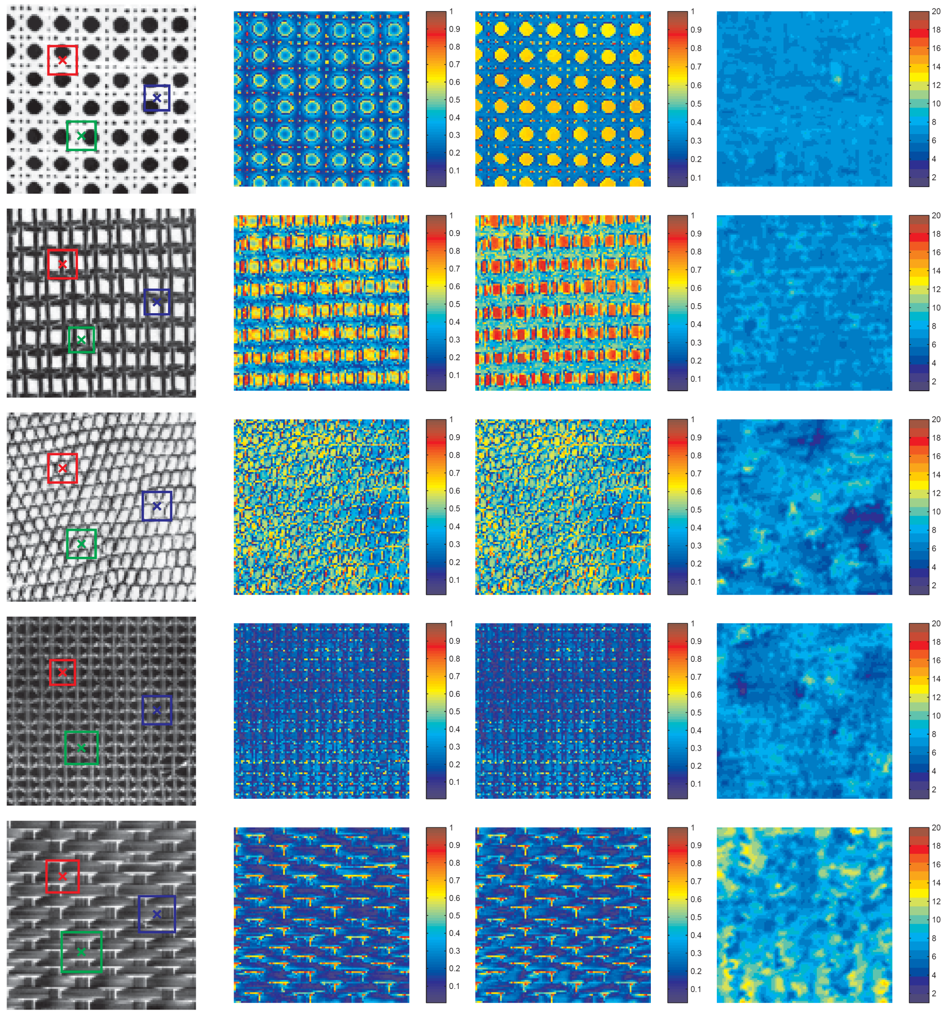
The solution of (2.19) is [56]:

$$v(x) = \max\{\min\{u(x) - \theta \ f(x), 1\}, 0\} \ . \tag{3.21}$$

The minimization scheme iterates (3.13), (3.14), (3.15), (3.16), (3.20), (3.19), and (3.21) alternately, until convergence. The discretization of div and ∇ are the same as described in [25, 56].

3.4 Experiments

We first show experimental results of the proposed scale model on several Brodatz textures. Fig.3.5(a) shows three arbitrarily chosen pixels on a Brodatz texture. In (b), the curve of entropy versus patch size at each indicated pixel is almost increasing and does not have a global maximum as patch size continues to increase. The histograms gain complexity as the patch size increases and there is no clear sign of the correct



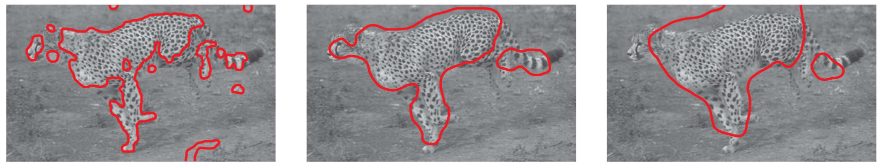
(a) Brodatz textures (b) Tikhonov (c) TV (d) Proposed

Figure 3.6: Comparison of scale maps with other methods on Brodatz texture images. (a) Brodatz texture images with patches with texton scales obtained by our model at arbitrarily selected pixels. (b) Scale map by Tikhonov flow. (c) Scale map by TV flow. (d) Scale map by the proposed model with $\alpha = 0.001$ and $\beta = 0.1$.

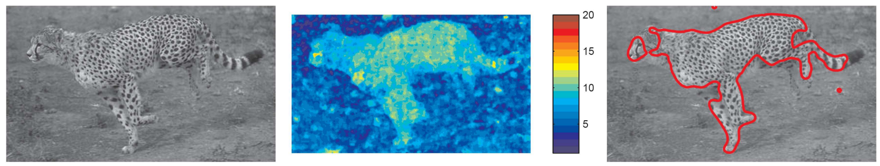
scale according to these curves, which emphasizes that entropy alone is not enough to find the scale. On the contrary, in (d), we see the histogram difference (using the Wasserstein distance) versus patch size obtains a global minimum and the texton size can be clearly identified at the first minimum from left, away from $r = 0$. In (c), the scale at each indicated pixel by the proposed model with the Wasserstein distance is accurate. In (f), the histogram difference versus patch size plot shows the Kullback-Leibler divergence captures the characteristics to some extend but not as well as the Wasserstein distance. The selected scale shown in (e) is roughly correct. The reason of the Wasserstein distance outperforming the Kullback-Leibler divergence in this experiment is that the Wasserstein distance overcomes the deficiency of pointwise metrics, as addressed in [54].

Fig.3.6 shows five Brodatz textures in column (a) and their scale maps by Tikhonov flow [49] in column (b), by TV flow [57, 50] in column (c), and by the proposed model in column (d). We use our own implementation of [57, 50, 49] in this experiment. The scale maps for these textures are expected to be homogeneous and only our model captures this characteristic. The parameters are $\alpha = 0.001$ and $\beta = 0.1$ in (3.9) for all five textures. We show in column (a) the scales obtained by our model at three arbitrarily selected locations which are accurate and agree with visual perception. In the first row, the scale map by Tikhonov flow highlights the edge of circles because the scale is associated with intensity gradients. The scale map by TV flow (d) highlights the circle regions since the scale is proportional to the size of a homogeneous region. Neither of the previous scale descriptors describe the size of the texture. We also apply the proposed scale model to several natural images from the Berkeley Segmentation Dataset as shown in the following.

Fig.3.7-3.9 shows the scale maps of the given images and compares the histogram based segmentation model [54] and the proposed model. (d) shows the given natural



(a) Scale 4 (b) Scale 16 (c) Scale 32



(d) Image (e) Scale map (f) Proposed model

Figure 3.7: (a) Segmentation with global scale $r = 4$. (b) Segmentation with global scale $r = 16$. (c) Segmentation with global scale $r = 32$. (d) Image. (e) Scale map by the proposed model. (f) Segmentation by proposed model.

images. (a), (b), and (c) are the segmentation results by [54] with global scale $r = 4, 16$, and 32, respectively. Fig.3.7(a), with $r = 4$, the segmentation selects within the cheetah patterns at some locations because the global scale is too small for those locations. In (b) (with $r = 16$) and in (c) (with $r = 32$), segmentation does not partition within the cheetah patterns but does not fall on the boundary accurately. This is because the global scale is too large, resulting many patches cross over both regions. The scale maps in (e) describe correctly each object region by a homogeneous scale and each background region by another homogeneous scale. The results in (f) by the proposed model with scale improve the segmentation results significantly.

3.5 Conclusion

In this work, we define a scale descriptor associated with texture. We propose a non-parametric model that seeks the scale by matching histograms in a self-repeating manner. The proposed energy functional consists of three terms. The first finds a size that satisfies a histogram matching criterion that compares the local patch with the neigh-

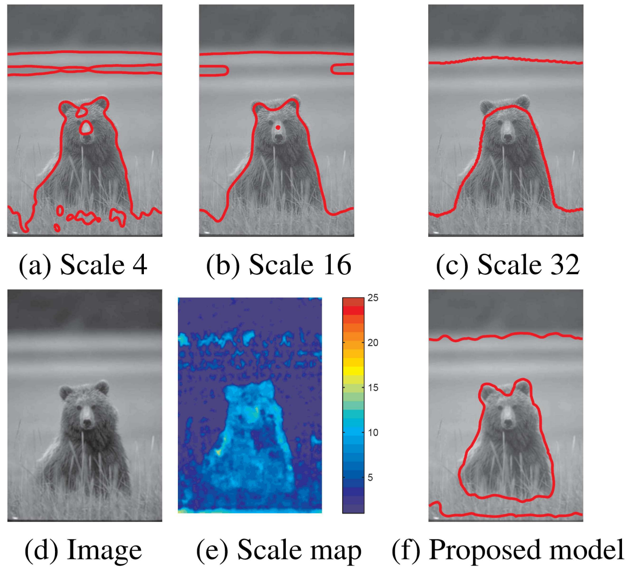


Figure 3.8: (a) Segmentation with global scale $r = 4$. (b) Segmentation with global scale $r = 16$. (c) Segmentation with global scale $r = 32$. (d) Image. (e) Scale map by the proposed model. (f) Segmentation by proposed model.

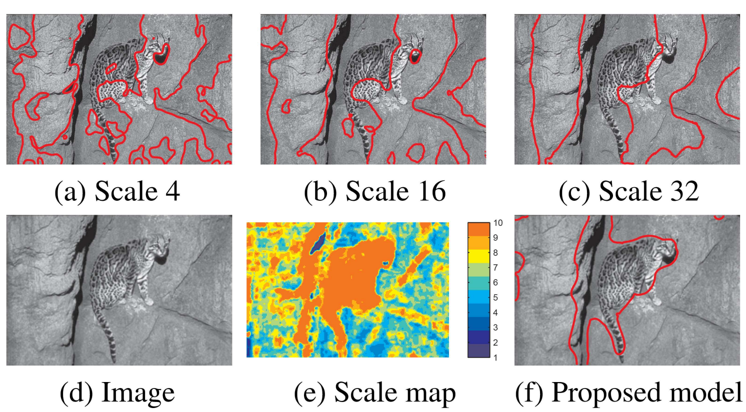


Figure 3.9: (a) Segmentation with global scale $r = 4$. (b) Segmentation with global scale $r = 16$. (c) Segmentation with global scale $r = 32$. (d) Image. (e) Scale map by the proposed model. (f) Segmentation by proposed model.

boring patch. The second maximizes the complexity of a patch to avoid choosing the wrong size when there are homogeneous regions within a texton. The third penalizes the size because texton is the smallest element that generates a texture. We show that these three terms are not redundant to one another. We also propose a modified model suited for finding the scale near texture edges. Our experimental results show that the scale map of a texture by the proposed model is highly accurate. Furthermore, we use scale as an image feature and also use it for characterizing local histograms in the proposed segmentation model. Our segmentation results on several natural images show an improvement over approaches that rely on a fixed scale.

CHAPTER 4

Unsupervised Multiphase Segmentation and Hierarchical Representation of Image Structure

4.1 Introduction

Image segmentation aims to partition an image domain into different regions in a meaningful way. Edge-based active contours methods [8, 10, 11] pose segmentation as an energy minimization problem and use edge detection functions that are based on local features to evolve contours towards object edges. Region-based active contours models incorporate both regions and edges to find a partition. Our proposed algorithm takes a region-based active contours approach because it is robust to noise and based on more global features. One of the early efforts towards region-based active contours was made by the Mumford and Shah segmentation model [12], which approximates a given image by a piecewise smooth image. However, the posed energy minimization problem is difficult to solve. Zhu and Yuille [14] use a family of Gaussian distributions to describe each region's data, i.e. mean and variance, and determine the boundaries of regions by competing with neighboring regions to best fit models at the largest possible areas. However, their proposed energy minimization problem is difficult to solve. Chan and Vese [1] is a two-phase piecewise constant model, a variant of Mumford and Shah. The novelty of their work is the use of the level set method to represent the evolving curve. The minimization is conveniently obtained by the gradient descent of the Euler-Lagrange equation of the energy functional. However, the extension of the Chan and Vese two-phase model to multiphase segmentation is not so natural. Several attempts have been made towards this extension. Vese and Chan [31] use n level set functions to represent up to 2^n regions because each level set function splits the image domain into two. This method implicitly represents the constraint of disjoint regions so no coupling forces are needed in order to constrain disjoint regions. However, n has to be properly chosen. Chung and Vese [58] use only one level set function but with level lines other than the zero-level line to represent contours. This method can represent n regions and the constraint of disjoint regions is also implicitly dealt with. However, their model cannot deal with triple junctions and the authors suggest combining their method with the Vese and Chan model to overcome this problem. Lie et al. [59] introduce to segmentation a piecewise constant level set function to represent each phase with a constant value. The piecewise constant constraint on the level set function is solved by using the augmented Lagrangian. Their level set method does not require re-initialization that is necessary for the classical level set method. However, extra work, as described in their paper, is needed for noisy images. The segmentation methods described above do not require any training set but the number of regions or at least an upper bound has to be given.

Brox and Weickert [60] use one level set function for each region to represent Zhu and Yuille's model. Brox and Weickert propose using a coupled curve evolution to solve this multiphase segmentation model but assume the number of regions is known. They also propose to automatically find the number of regions by a coarse-to-fine strategy coupled with a hierarchical splitting. The authors apply a two-phase segmentation on a subregion, and if the Zhu and Yuille's energy functional is lowered, they continue the current segmentation. This process is repeated for all regions until the Zhu and Yuille's energy functional cannot be lowered. The number of phases obtained by this procedure is used in their multiphase segmentation model. Sandberg et al.'s segmentation model [61] automatically determines the number of regions and finds partitioning simultaneously. In the energy functional, they introduce a feature balancing term, the sum of all the inverse of region scales from each region, which is used to implicitly penalize the number of regions in addition to the total length of the boundaries. The *region scale* of a region is the quotient of its area and perimeter. Their model can be easily solved by a pixel-wise decision algorithm which implicitly deals with the disjoint constraint on all phases. This minimization method is very efficient but not robust to noise.

In this chapter, we provide a spatial enclosure relationship between higher-level and lower-level regions so that one can analyze an image at a certain level of scale. *Scale* is related to contrast and region scale, and we use the definition of the TV scale in [57], which is defined as the time taken for a feature to disappear under the total variation flow. Tu and Zhu [32] consider segmentation a computing process rather than a vision task. The more one looks at an image, the more one sees. Therefore, segmentation results are not universal. We provide a "structure" of an image because that is how an image is usually interpreted. Following this idea, we propose to start from the coarsest partitioning and then refine each partitioning individually. Our proposed multiphase piecewise constant segmentation first applies the Chan and Vese model to partition an image domain into two and then recursively applies the Chan and Vese model in each partitioned region. This procedure gives a structure of an image implicitly utilizing the notion of "saliency" [62] that involves scale and intensity contrast in its determination. We additionally propose some stopping conditions to terminate the two-phase segmentation on the indicated region when it becomes meaningless to partition further. The stopping conditions use region scale and contrast to detect over-segmentation.

4.2 Two-phase Piecewise Constant Segmentation on an Indicated Region

In this section, we first describe previous two-phase piecewise constant segmentation models and then present a natural extension to partition any given subregions that may be of arbitrary shapes. Let $f : \Omega \to [0, L]$ be the given grey-scale image. A two-phase piecewise constant version of the Mumford-Shah model [12] evolves a curve C towards the boundary between two regions and approximates f by two constants c_1 and c_2 inside the curve C and outside the curve C, respectively. The Chan and Vese model [1] is the following energy minimization problem:

$$\inf_{C,c_1,c_2} \left\{ E^1[C, c_1, c_2] = \int_C ds \quad + \quad \lambda \int_{inside(C)} (c_1 - f(x))^2 \, dx \right. \\ \left. + \lambda \int_{outside(C)} (c_2 - f(x))^2 \, dx \right\}, \tag{4.1}$$

where the first term measures the total length of the curve C to penalize complicated interface between two regions and λ is a scalar parameter that controls the balance between regularization and data. This model can be represented in the following level-set formulation:

$$\inf_{\phi,c_1,c_2} \left\{ E^1[\phi, c_1, c_2] = \int |\nabla H(\phi(x))| \quad dx \quad + \lambda \int H(\phi(x))(c_1 - f(x))^2 \, dx \right. \\ \left. + \lambda \int \Big[1 - H(\phi(x))\Big] (c_2 - f(x))^2 \, dx \right\}, \tag{4.2}$$

where H is the Heaviside function and ϕ is the level set function [13] such that $\phi >$

0 inside C and $\phi < 0$ outside C. The minimization of this level set formulation can be solved naturally by the standard PDE method [1] and allows topological changes of the curve. However, this model is not convex and thus a reasonable initialization is necessary to avoid getting stuck at undesired local minima.

Bresson et al. [5] proposed a fast global minimization of the Chan and Vese model. There are two major advantages of their algorithm. The first is that the initialization can be arbitrary. The second is that the solutions can be obtained much faster than the standard PDE method. Bresson et al.'s model is the following minimization problem:

$$\min_{u,0\leq v\leq 1,c_1,c_2} \left\{ E^2_\Omega[u,v,c_1,c_2] = TV_\Omega(u) + \frac{1}{2\theta}||u-v||_{L^2(\Omega)} \right. \tag{4.3}$$
$$\left. +\lambda \int_\Omega v(x)(c_1 - f(x))^2 + [1-v(x)](c_2-f(x))^2 \, dx \right\},$$

where θ is small enough so that u and v are are significantly close, λ is a parameter controlling the data fidelity term, and the total variation of u is defined in the following:

$$TV_\Omega(u) = \sup \left\{ \int_\Omega u \text{ div } p \, dx \;\middle|\; p \in C^1_c(\Omega; \mathbb{R}^2) : |p(x)| \leq 1, \forall x \in \Omega \right\}. \tag{4.4}$$

If $u^* =$ argmin $E^2_\Omega[u,v,c_1,c_2]$, the partition can be chosen to be, for instance, $\{u^* \geq 0.5\}$ and $\{u^* < 0.5\}$. Let $r(x,c_1,c_2) = (c_1 - f(x))^2 - (c_2 - f(x))^2$. The minimization is solved by alternating the following equations [5]:

$$c_1 = \frac{\int_\Omega f(x)v(x)dx}{\int_\Omega v(x)dx} \tag{4.5}$$

$$c_2 = \frac{\int_\Omega f(x)[1-v(x)]dx}{\int_\Omega [1-v(x)]dx} \tag{4.6}$$

$$p(x) = \frac{p(x) + \Delta t \; \nabla(\text{div } p(x) - v(x)/\theta)}{1 + \Delta t|\nabla(\text{div } p(x) - v(x)/\theta)|} \tag{4.7}$$

$$u = v - \theta \text{ div } p \tag{4.8}$$

$$v(x) = \min \Big\{ \max \Big\{ u(x) - \theta\lambda r(x, c_1, c_2), 0 \Big\}, 1 \Big\} \ , \tag{4.9}$$

where Δt is the time step. These equations are iterated until convergence.

Let the resolution of image be $M \times N$ and write $p = (p^1, p^2)$. The discretization of div and ∇ that satisfy the definition of TV norm in equation (4.4) are defined in the following ways [5, 25]:

$$\Big(\nabla u\Big)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < M \\ 0 & \text{if } i = M \end{cases} \tag{4.10}$$

$$\Big(\nabla u\Big)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases} \tag{4.11}$$

$$\Big(\text{div } p\Big)_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < M \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = M \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = N \ . \end{cases} \tag{4.12}$$

We generalize Bresson et al.'s algorithm described above for partitioning a given region $S \subseteq \Omega$ that may have arbitrary shapes, in the following:

$$\min_{u, 0 \le v \le 1, c_1, c_2} \Bigg\{ E^2_S[u, v, c_1, c_2] = TV_S(u) + \frac{1}{2\theta} ||u - v||_{L^2(S)} \tag{4.13}$$

$$+\lambda \int_S v(x)(c_1 - f(x))^2 + [1 - v(x)](c_2 - f(x))^2 \, dx \Bigg\} \ ,$$

The minimization equations are the same as equations (4.5)-(4.9), except the solutions are restricted on the indicated region S. Simple modifications of (4.10), (4.11), and (4.12) can be made to suit arbitrary regions S and still satisfy the definition of the TV norm in equation (4.4). For all (i,j) such that $\chi_S(i,j) = 1$,

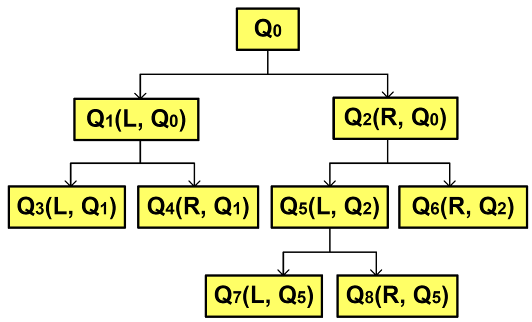
$$\left(\nabla u\right)^1_{i,j} = \chi_S(i+1,j)(u_{i+1,j} - u_{i,j}) \tag{4.14}$$

$$\left(\nabla u\right)^2_{i,j} = \chi_S(i,j+1)(u_{i,j+1} - u_{i,j}) \tag{4.15}$$

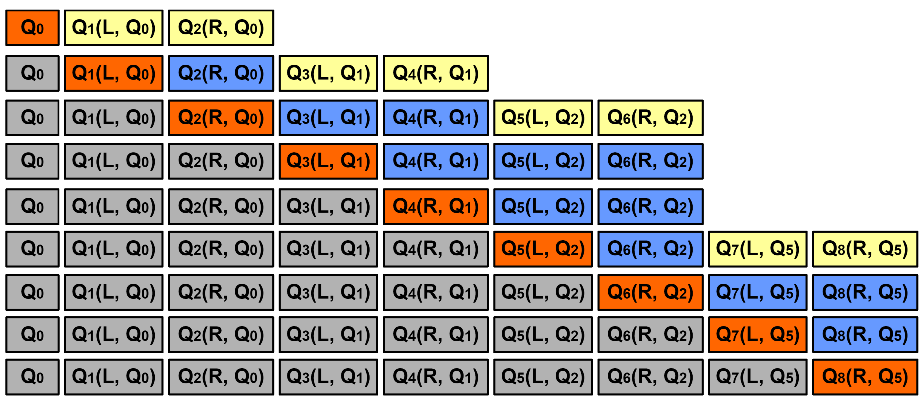
$$\begin{aligned} \left(\text{div } p\right)_{i,j} \quad = \quad & \chi_S(i+1,j)\, p^1_{i,j} - \chi_S(i-1,j)\, p^1_{i-1,j} \\ & + \chi_S(i,j+1)\, p^1_{i,j} - \chi_S(i,j-1)\, p^1_{i,j-1}\ . \end{aligned} \tag{4.16}$$

4.3 Hierarchical Representation of an Image

In this section, we present our proposed segmentation algorithm which provides a full hierarchical representation of the structure of a given image. The proposed algorithm recursively applies equation (4.13) to split a partitioned region into two and generates an ordered binary tree to represent the structure of the image. Fig. 4.1(a) is an example of the process of the proposed recursive segmentation. Initially, the two-phase piecewise constant segmentation is applied on the entire image domain so the root node, Q_0, represents the entire image domain. The partitioned regions of Q_0 are stored in $Q_1(L, Q_0)$ and $Q_2(R, Q_0)$ where the second place, Q_0, indicates the parent node, and L and R represent left child node and right child node, respectively. Since the initialization of the two-phase segmentation (4.13) can be arbitrary, we choose the



(a) Structure of an image represented in an ordered binary tree



(b) Proposed algorithm

Figure 4.1: An example of the proposed recursive two-phase segmentation. (a) shows an ordered binary tree that represents the structure of an image. (b) shows the process, from top to bottom, of the recursive segmentation in which each partitioned region is stored in a queue. The bottom row is the final tree structure in the queue representation.

image itself (normalized to range from 0 to 1) as initialization for v in (4.13). The segmented regions are $\{u \geq th\}$ and $\{u < th\}$, where th is the intensity mean of the current region (or the union of both segmented regions). The left child node represents region $\{u \geq th\}$ and the right child represents region $\{u < th\}$. Since the minimization is a gradient descent, by our initialization, the intensity mean of the left child node is always higher than that of the right child node. In this way, the order of intensity means of the partitioned regions are preserved and we obtain an ordered binary tree. Next, we apply the two-phase segmentation on the region in $Q_1(L, Q_0)$ and store the segmented regions into $Q_3(L, Q_1)$ and $Q_4(R, Q_1)$. Then, we continue this method and proceed segmentation on $Q_2(L, Q_0)$, which is split into $Q_5(L, Q_2)$ and $Q_6(R, Q_2)$. This process gives us a hierarchical representation of the image structure. However, this process becomes meaningless when the scale of structure becomes too small. Therefore, we propose to terminate segmentation of a region if one of the following three conditions is satisfied. The first is when the evolving curve disappears, which happens naturally if the current region has homogeneous intensity. The second stopping condition uses region scale to prevent oversegmentation. The third is when the approximated constants are so similar that it becomes meaningless to partition that region. In this fashion, a region that satisfies any of the stopping condition above becomes a leaf of the ordered binary tree. In the example of Fig. 4.1, region $Q_3(L, Q_1)$ meets one of the stopping criteria and thus has no child nodes. Similarly, region $Q_4(R, Q_1)$ has no child nodes.

To implement the recursive segmentation described above, we use a queue to store the segmented regions. Fig. 4.1(b) shows the procedure, from top to bottom, of the implementation of the example in Fig.4.1(a). The first row shows that the current node is Q_0 and the partitioned regions are stored into $Q_1(L, Q_0)$ and $Q_2(R, Q_0)$. The subindex of Q is the position of entity in the queue. The second row shows that the current node is $Q_1(L, Q_0)$ and the partitioned regions are stored into $Q_3(L, Q_1)$ and

$Q_4(R, Q_1)$. The third row shows that the current node is $Q_2(R, Q_0)$ and the partitioned regions are stored into $Q_5(L, Q_2)$ and $Q_6(R, Q_2)$. The forth row shows that the current node is $Q_3(L, Q_1)$ and does not need to be partitioned. This process is continued until there is no more node stored in the queue. The pseudo code is shown in Algorithm 1. Initially, the region to be segmented is the entire image domain and this is represented by a characteristic function χ_Ω. 'Phase' is a matrix of the size of image that records the leaf regions and j is the label of each phase; n keeps track of current entity in queue to be segmented, and N is the position of the entity in queue a newly segmented region to be stored. The function 'two-phase piecewise constant segmentation' uses equation (4.13) and returns two characteristic functions χ_1 and χ_2 that represent the segmented regions on the current region. It additionally returns a true and false variable 'split' according to the stopping conditions. If one of the stopping conditions is satisfied, 'split' is false; otherwise, 'split' is true.

Algorithm 1 Recursive two-phase segmentation

Given image $f : \Omega \to [0, L]$
$Q_0 \Leftarrow \chi_\Omega$
Phase $\Leftarrow \vec{0}$
$n \Leftarrow 0$
$N \Leftarrow 0$
$j \Leftarrow 0$
while Q_n exists **do**
  $[\chi_1, \chi_2$, split] = two-phase piecewise constant segmentation(f, Q_n)
  if split = true **then**
    $Q_{N+1}(L, Q_n) \Leftarrow \chi_1$
    $Q_{N+2}(R, Q_n) \Leftarrow \chi_2$
    $N \Leftarrow N + 2$
  else
    Phase $\Leftarrow$ Phase + $j * Q_n$
    $j \Leftarrow j + 1$
  end if
  $n \Leftarrow n + 1$
end while

4.4 Experimental results

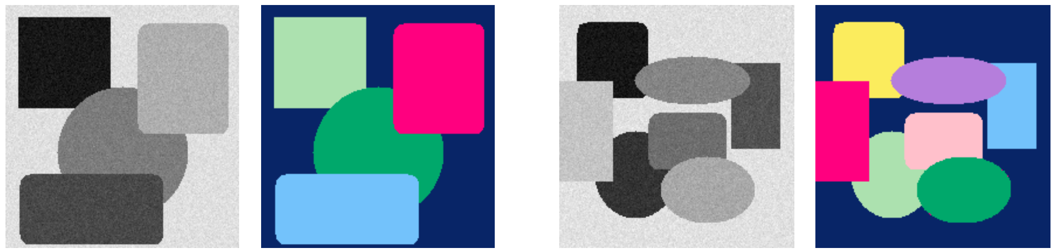
Fig. 4.2 shows experiments on synthetic images with five and eight regions, respectively. The proposed algorithm automatically detects each region of the images. The segmentation results are shown in different colors to represent each phase of the segmentation. As can be seen, the recursive approach implicitly deals with junctions. Fig. 4.3 shows the ordered binary tree structure of the image. The red region represents the extracted region, black represents the irrelevant region, and blue represents the complement of the red in the relevant region. Fig. 4.4 shows the binary tree structures of image (a) and image (b) in the queue representation. We will show other results of image structures in the queue representation.

In Fig. 4.5, (a) shows that the proposed segmentation starts from the coarsest scale. The two rectangles are first separated from the background and then separated from each other. In (b), the contrast of the right rectangle is high and also the region scale of it is high enough. Therefore, the right rectangle is first separated from the union of the left rectangle and background, and then the left rectangle and background are separated. This is the case when the contrast is high enough to have greater influence on the segmentation result than the region scale. Fig. 4.6 also shows that recursive segmentation competes between the region scale and contrast in a reasonable manner. In (a), the contrasts from the background to the top-left rectangle, from the top-left to the top-right, from the top-right to the bottom-left, and from the bottom-left to the bottom-right are all the same. All five regions have the same area but the perimeter of the background is larger than those of the rectangles. Therefore, as expected according to region scale, the first segmentation separates the rectangle with lowest intensity and the background from the rest of the rectangles. In (b), the bottom-right rectangle has the highest contrast and its region scale is large enough. Therefore, the first segmentation extracts it from other regions. Fig. 4.7 shows that the order of segmentation is from the largest region scale, as in (a) and (d), and also depends on the contrast, as in (b) and (c). In (a), the center circle has a very high contrast but its region scale is small. Therefore, all the circles as a unit are separated from the background in the first segmentation step. In (b), (c), and (d), all high-contrasted circles' scales are large enough and thus the segmentation is performed according to the contrast. All results show that recursive segmentation competes region scale and contrast in an intuitive way.

Fig. 4.8 and 4.10 show the segmentation results of some real images. The proposed algorithm automatically detects each region of the given images. The segmentation results are obtained by applying the proposed algorithm on the respective grey-scale images of the color images shown in the figure. The proposed algorithm can be naturally applied to vector images such as color images. Fig. 4.9 and 4.11 show that the images are first roughly partitioned into two regions of similar sizes and then partitioned at finer scale as the tree level goes down. In fig. 4.9(a), the sky and far mountains are separated from the near mountains and trees. Then, the sky and the far mountains are separated and the near mountains and trees are separated. Finally, different layers of the near mountains are separated. In (b), the sky and far mountains are separated from near mountains and rocks. Then, the sky and far mountains are separated and the near mountains and rocks are separated. In 4.11(a), the trees and sky are separated first. Then, the sky is partitioned into two because its intensity is not homogeneous. Finally, the moon is separated from the sky. In (b), the sky and the buildings are separated first, except for the roof because its intensity is similar to that of the sky. Then, the roof is separated from the sky.

4.5 Conclusion

In this chapter, we proposed a segmentation algorithm that provides the structure of an image from the largest scale. The proposed algorithm recursively applies Bres-



given image　segmentation result　given image　segmentation result

Figure 4.2: The proposed algorithm automatically detects each region in the given images and junctions are preserved. Each color represents a phase of the segmentation.

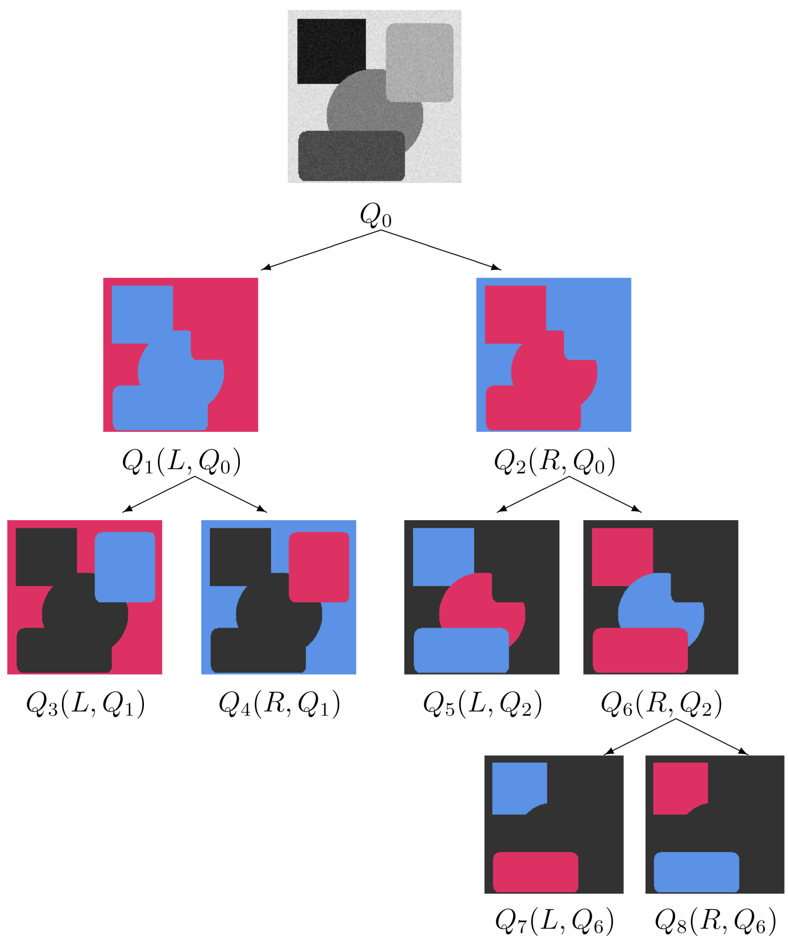


Figure 4.3: Image structure by the proposed segmentation. Red: extracted region, black: irrelevant region, blue: complement of red in relevant region.

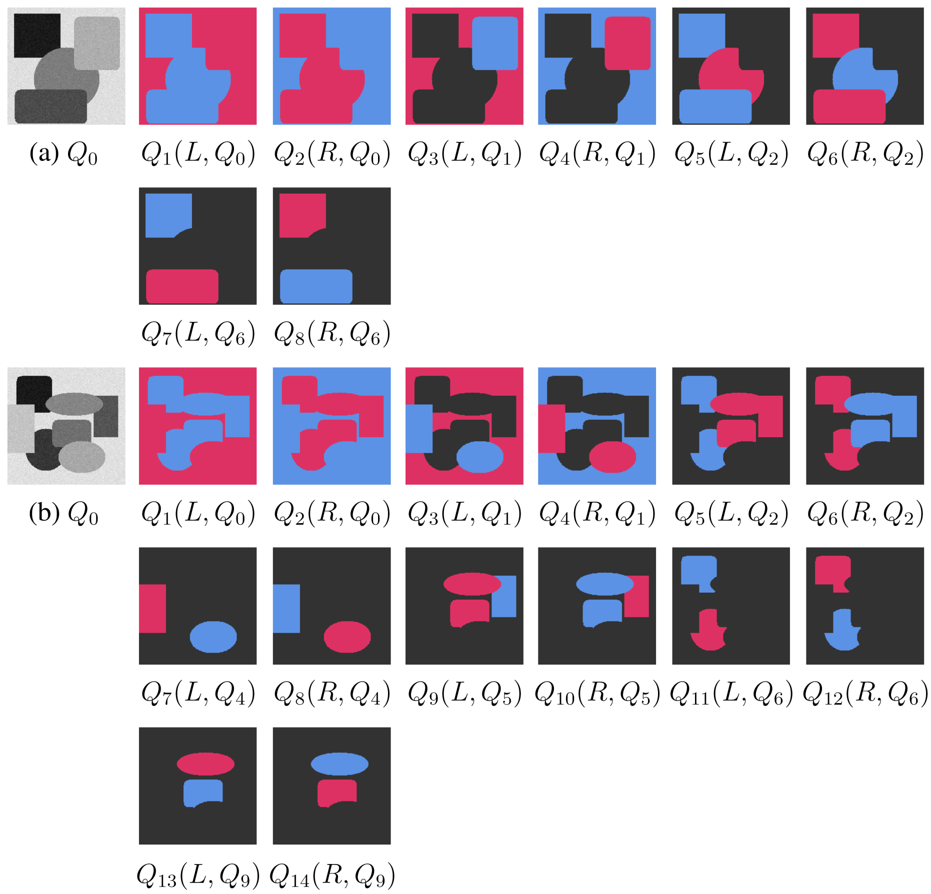


Figure 4.4: Tree structures of image (a) and image (b) in the queue representation.

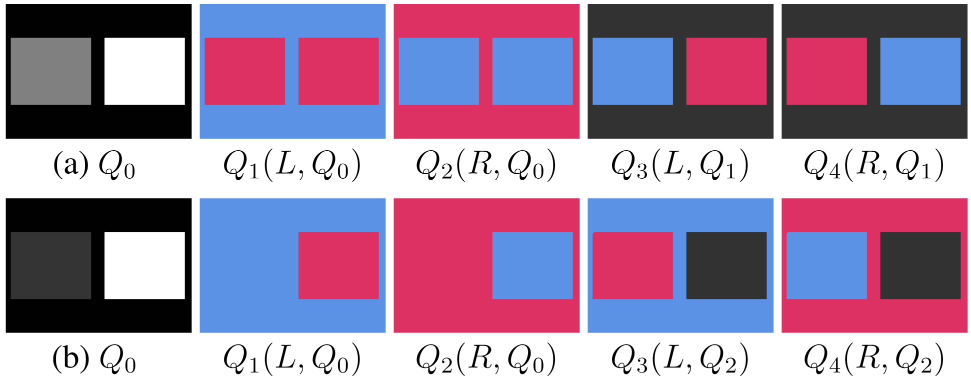


Figure 4.5: (a) shows that segmentation starts from the largest region scale. In (b), the right rectangle has a higher contrast and its region scale is also large enough, so segmentation first separates it from the rest.

son et al.'s two-phase piecewise constant segmentation on the partitioned region. We defined an ordered binary tree to represent this process to show the structure of an image. This ordered binary tree is implemented by storing each node as an entity in a queue. We proposed three stopping conditions to detect when segmenting a region further is trivial. This can be used to give a multiphase segmentation result, in which the leaf of the tree represents each phase of the multiphase segmentation. Given the stopping conditions, the number of phases can be unknown and can be arbitrary. Additionally, junctions of contours are implicitly dealt with by the recursive segmentation method. We have shown several experiments on synthetic images to see that the proposed segmentation follows the order of region scale and contrast in a reasonable manner. Experimental results on real images show that the obtained image structures are consistently intuitive.

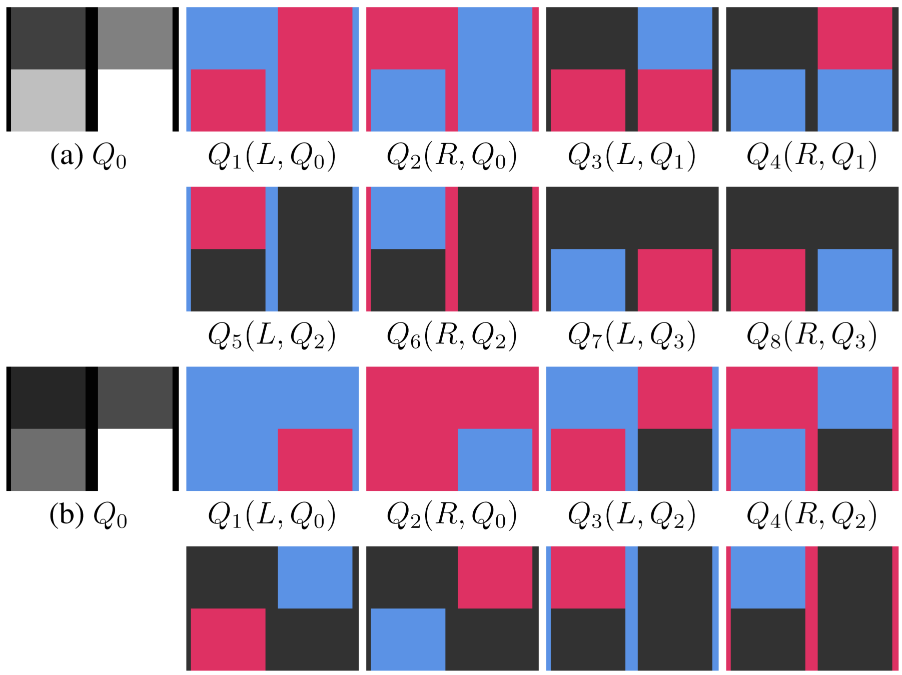


Figure 4.6: The proposed segmentation starts from the largest region scale and highest contrast. In (a), the contrasts from the background to the top-left rectangle, from the top-left to the top-right, from the top-right to the bottom-left, and from the bottom-left to the bottom-right are all the same. All five regions have the same area but the perimeter of the background is larger than those of the rectangles. Therefore, the first segmentation separates the rectangles with the lowest intensity and the background from the rest. (b) The bottom-right rectangle has the highest contrast and is extracted first.

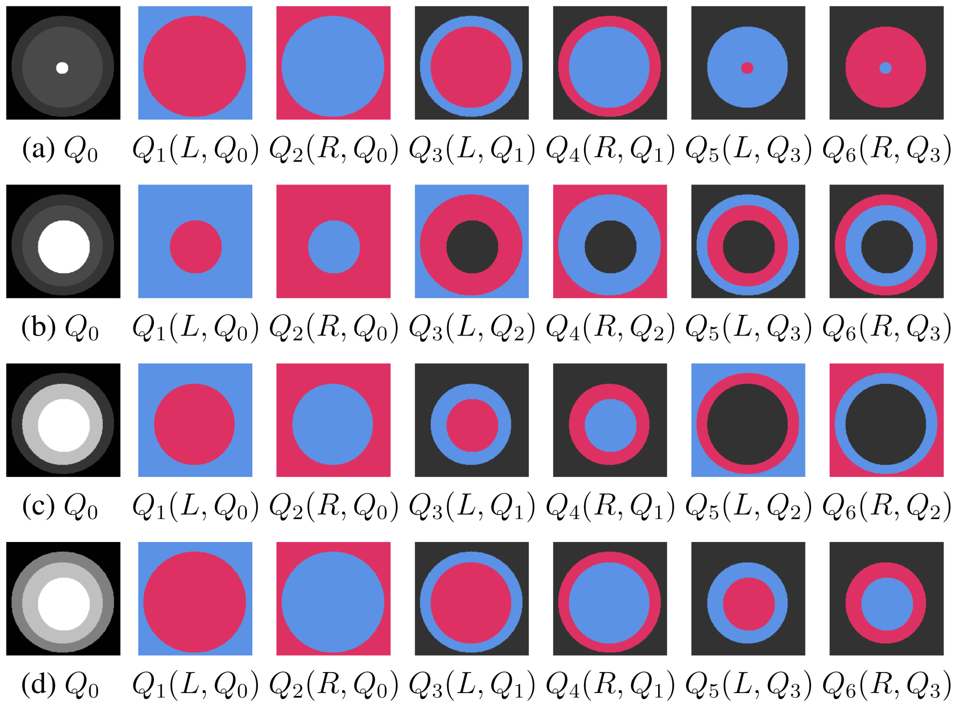


Figure 4.7: The segmentation is from the largest region scale, in (a) and (d), but also depends on the contrast when region scale is large enough, in (b) and (c).

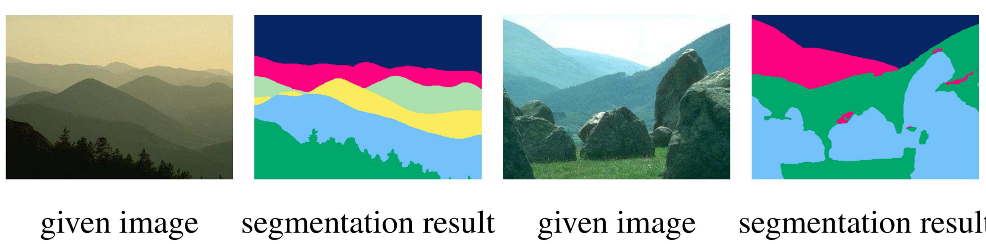


Figure 4.8: The proposed algorithm automatically detects each region in the given images. The segmentation results are obtained by experimenting on the respective grey-scale images of the color images shown here.

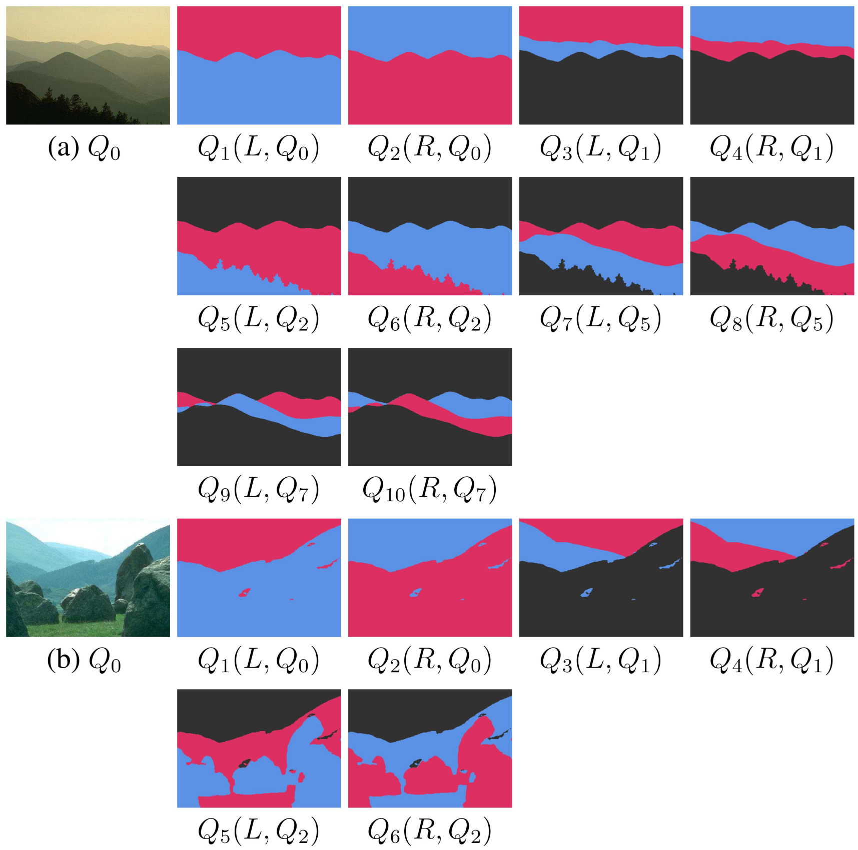


Figure 4.9: The structures of the images in the queue representation. These images are first roughly partitioned into two regions of similar sizes and then partitioned at finer scale as the tree level goes down. (a) First the sky and far mountains are separated from the near mountains and trees. (b) First the sky and far mountains are separated from near mountains and rocks.

given image segmentation result given image segmentation result

Figure 4.10: The proposed algorithm automatically detects each region in the given images. The segmentation results are obtained by experimenting on the respective grey-scale images of the color images shown here.

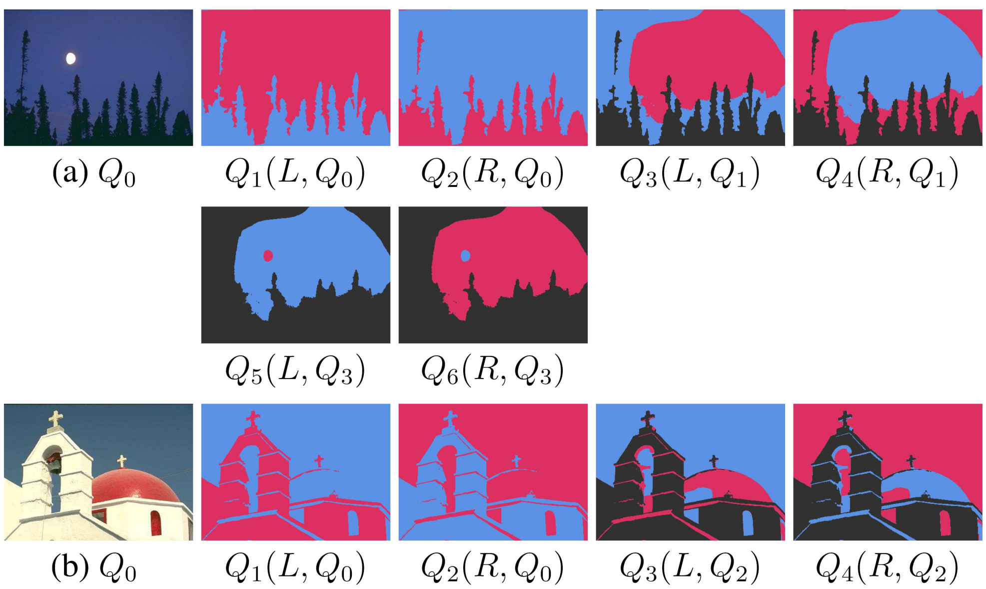


Figure 4.11: The structures of the images in the queue representation. These images are first roughly partitioned into two regions of similar sizes and then partitioned at finer scale as the tree level goes down. (a) The trees and sky are separated first. (b) The sky and the buildings are separated first, except for the roof because its intensity is close to that of the sky. The experiments are on the respective grey-scale images of the original color images shown here.

CHAPTER 5

Texture and Geometric Inpainting

5.1 Introduction

Inpainting is an image interpolation problem or the task of repairing the damaged region of an image. For example, an artist fills in an old painting's missing pieces by using the colors from the surrounding areas. Digital inpainting provides a safe method to experiment with different colors without damaging the original painting. Moreover, with the increase of digital photography, digital inpainting plays an important role in image processing and also the movie industry when creating special effects, such as removing unwanted objects in an image. In this chapter, we focus on automatic inpainting techniques rather than sophisticated graphics software that require tremendous manual labor. Once the inpainting region is provided by the user, the inpainting process is automatic.

Bertalmio et al. [63] introduce PDE inpainting models into the field of image processing. If I is the given image intensity, Ω is the image domain, and $D \in \Omega$ is the inpainting region, the image is repaired by propagating I along its isophotes' direction into D by the following PDE

$$\frac{\partial u}{\partial t} = \nabla^{\perp} u \cdot \nabla \triangle u, \tag{5.1}$$

where $\nabla^{\perp} = (-\partial_y, \partial_x)$ and u denotes the repaired intensity. When u reaches a steady state, it is a solution that repairs the original image I in the region D. Additional anisotropic diffusion is applied to u once every few iterations of (5.1) to preserve sharpness of edges, as shown in the following equation

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) |\nabla u|. \tag{5.2}$$

Bertalmio's subsequent work with Bertozzi and Sapiro [64] makes the connection with classical fluid dynamics. In particular, their method is directly based on the two dimensional Navier-Stokes equations. The image intensity, u, is analogous to the stream function of fluid; isophote direction, $v = \nabla^{\perp} u$, is analogous to fluid velocity; and smoothness of image $w = \triangle u$ is analogous to vorticity. Instead of solving the transport equation described in (5.1), they solve a vorticity transport equation for w,

$$\frac{\partial w}{\partial t} + v \cdot \nabla w = \mu \nabla \cdot (g(|\nabla w|) \nabla w), \tag{5.3}$$

where μ is a scalar parameter and $g(|\nabla w|)$ allows anisotropic diffusion. The repaired image, u, is then obtained by solving

$$\triangle u = w,$$

with boundary condition

$$u|_{\partial D} = I.$$

A crucial observation of the inpainting problem is that sharp edges play an important role in visual perception. Inspired by Bertalmio et al.'s [63] PDE approach, Chan and Shen [65] propose a TV (total variation) inpainting model based on the Bayesian and variational principles. TV inpainting is adapted from the Rudin-Osher-Fatemi denoising model [52] and is an edge-based model that minimizes the total variation of an image over the inpainting region with suitable boundary conditions. The image is repaired by the following energy minimization

$$\min_{u \in BV(\Omega)} E_{tv}[u] = \int_{\Omega} |\nabla u|, \tag{5.4}$$

with constraint

$$u \mid_{\Omega - D} = I \mid_{\Omega - D} . \tag{5.5}$$

Minimizing this energy functional is equivalent to connecting the level sets across the inpainting region with the smallest distance. This can be seen by the coarea formula

$$\int |\nabla u| dx = \int_0^1 \int_{\Gamma_\lambda} ds d\lambda \ ,$$

where $\Gamma_\lambda = \{x : u(x) = \lambda\}$ is the level set and ds is the arc length of the level sets. The total variation of u is the total length of all level sets. Therefore, with the constraint in (5.5), sharp edges are connected according to the level lines in the known region.

The gradient descent of the Euler-Lagrange equation for (5.4) is

$$\frac{\partial u}{\partial t} = \nabla \cdot (\frac{\nabla u}{|\nabla u|}), \tag{5.6}$$

with constraint

$$u \mid_{\Omega - D} = I \mid_{\Omega - D} .$$

TV inpainting interpolates images across the missing regions, while preserving sharp edges. However, it does not always connect edges correctly. For example, when the inpainting region is too large, the level lines may form kinks at the boundary of the inpainting region in order to make the shortest level line connection, which often does not agree with visual perception.

Chan, Kang, and Shen [6] proposed to minimize Euler's elastica energy instead of the total variation. Euler's elastica inpainting improves TV inpainting because the curvature of level lines is additionally penalized. Therefore, it does not allow kinks at the inpainting region boundary because the curvature of kinks is infinite. Moreover, curved edges, as well as straight edges, are extended correctly across the inpainting region. Euler's elastica inpainting repairs the missing region by the following energy minimization:

$$\min_{u \in BV(\Omega)} E_{elas}[u] = \int (a + b\kappa^2)|\nabla u| dx, \tag{5.7}$$

with constraint

$$u \mid_{\Omega - D} = I \mid_{\Omega - D},$$

leaving the outside of the inpainting region untouched. In this functional, a and b are positive constants and $\kappa = \nabla \cdot (\frac{\nabla u}{|\nabla u|})$ is the curvature of u. Minimizing this energy functional is equivalent to connecting sharp edges according to the curvature of the level lines in the known region. This can be explained by the coarea formula

$$\int (a + b\kappa^2)|\nabla u| dx = \int_0^1 \int_{\Gamma_\lambda} (a + b\kappa^2) ds d\lambda \ .$$

The gradient descent of the Euler-Lagrange equation of model (5.7) is

$$\frac{\partial u}{\partial t} = \nabla \cdot \left[(a + b\kappa^2) \overrightarrow{n} - \frac{\overrightarrow{t}}{|\nabla u|} \frac{\partial (2b\kappa |\nabla u|)}{\partial \overrightarrow{t}} \right], \tag{5.8}$$

where $\overrightarrow{n} = \frac{\nabla u}{|\nabla u|}$ is the normal vector, $\overrightarrow{t} = \overrightarrow{n}^{\perp}$ is the tangent vector, and $\frac{\partial}{\partial \overrightarrow{t}} = \overrightarrow{t} \cdot \nabla$ is the directional derivative.

There are additional methods that involves solving stiff fourth order PDEs in the inpainting problem. Bertozzi et al. [66, 67] proposed a model for inpainting binary images by modifying the Cahn-Hilliard equation. They made the observation that the fourth order gradient flow of the Esedoglu-Shen's inpainting model [68], based on the Mumford-Shah-Euler image model, has partial commonality with the Cahn-Hilliard equation, for which fast solutions are available. The two-scale Cahn-Hilliard model for inpainting is successful in connecting edges across large inpainting regions, at orders of magnitude faster than Euler's elastica inpainting. Recently, Dobrosotskaya and Bertozzi [69] proposed a wavelet-based inpainting which adapts the second-order Allen-Cahn equation and maintains comparable inpainting speed. Additionally, their model was extended to grey-scale images.

There are various works that combine PDE-based inpainting and texture synthesis. In [70], an image is first decomposed into the sum of two functions including a geometric part and a texture part. The inpainting region of the geometric part is repaired by [63]'s PDE inpainting, while the texture part is repaired by texture synthesis [7]. The final solution is the sum of the repaired parts. Criminisi et al. [71] improve exemplar-based texture synthesis by using a PDE-inspired method to determine the order of the filling process. The inpainting algorithm by Grossauer [72] also decomposes an image into a geometric part and a texture part. The geometric part is repaired by a PDE inpainting method and then is segmented into different regions. Texture synthesis is performed on the texture, on which each of the segmented regions in the inpainting region are synthesized separately.

In this chapter, we propose a new inpainting algorithm that works for both textured and non-textured areas of an image. As described above, [6] introduced a technique for inpainting using an Euler's elastica energy-based variational model that works well for repairing smooth areas of the image while maintaining edge detail. However, their technique is very slow due to a stiff, fourth order PDE. One the other hand, Efros and Leung [7] used texture synthesis techniques for inpainting and hole filling. This works well for areas of an image that contain repeating patterns. Demanet et al. [73] proposed a correspondence map formulation for inpainting that is related to texture synthesis methods in [7, 74]. Many other works relating to texture synthesis exist in the literature, although this is not the focus of this work, and will not be presented. In the next section, we present Efros and Leung's texture synthesis. We use the idea of texture synthesis to accelerate and constrain the PDE solution of elastica inpainting. Instead of a stiff minimization, we have a combinatorial optimization problem. This method gets close to the PDE solutions of elastica inpainting. We further combine this algorithm with texture synthesis by thresholding so that repeating patterns are fully utilized.

5.2 Non-parametric Texture Synthesis

Texture synthesis by non-parametric sampling in Efros and Leung's work [75] synthesizes texture using an initial seed. This method can be used for filling in missing image data when the image has repeating patterns and the undamaged part can be used as an initial seed. Texture synthesis models texture as a Markov Random Field and uses a pixel's local patch as conditional probability to search for similar patches. It grows pixels one at a time, starting from the edge of the initial seed. For inpainting, the missing region is repaired from its boundary inward, in concentric layers. To do this, each pixel is initially assigned with a confidence value from 0 to 1, 1 being known and 0 being unknown. Then, compute the priority map, which is the sum of the confidence values of a pixel's eight neighboring pixels. The priority value of a missing pixel represents the count of the known pixels surrounding it. The order in which the pixels are repaired follows the priority values from high to low. The confidence map and priority map are updated every time a pixel is repaired, because a repaired pixel becomes partially known. To repair a pixel, its local patch colors are used to search for similar patches in the undamaged part. The degree of similarity between two patches is measured by the Sum of Squared Differences (SSD) on the patch colors. In addition, the SSD is weighted in favor of pixels near the center pixel. Finally, the color of the center pixel from the most similar patch is copied into the current pixel.

We summarize one iteration of the texture inpainting algorithm in the following steps:

1. Update the confidence map and the priority map.
2. For the remaining missing pixels, find one with the highest priority.
3. Fill in the current pixel using its local patch and search for a similar patch according to the weighted SSD.

These steps are iterated until a steady state is reached.

5.3 Constrained Elastica Inpainting Algorithm

The minimization flow, Eq. (5.8), that solves the elastica inpainting problem is a stiff fourth-order PDE. To avoid stiff minimization, we propose a fast algorithm that approximates the solution of Euler's elastica inpainting. The proposed algorithm is based on the observation that the colors of missing pixels are likely to exist in the image. Using the idea of texture synthesis, some of the pixels neighboring an unknown pixel can be used if they are known. Therefore, we have a constrained elastica inpainting problem, which turns the stiff minimization problem into a combinatorial optimization problem. To fill in a missing pixel, the neighboring pixels are marked "known", and these are used to find the best color candidate according to the Euler's elastica energy.

Like texture synthesis, the algorithm proceeds from the boundary of the inpainting region inward in an upwind scheme. We use also the confidence map, in which a pixel's confidence value equals the number of known neighboring pixels, to proceed this ordering. The missing pixel is replaced by the candidate that minimizes Euler's elastica energy of the local neighborhood. This process is iterated until pixel colors no longer change or a steady-state solution is reached. We summarize one iteration of the proposed algorithm in the following steps:

1. Update the confidence map and the priority map.
2. For the remaining missing pixels, find one with the highest priority.
3. Using a greedy algorithm, choose the color, from all color candidates in the image, that minimizes Euler's elastica energy of the local area, while the neighboring colors are fixed.

These steps are iterated until a steady state is reached. This algorithm approximates the solution of elastica inpainting and converges very fast compared to the PDE solver (5.8). Experiments show that the proposed algorithm is two orders of magnitude faster [76].

5.4 Texture and Geometric Inpainting Algorithm

The proposed algorithm in the previous section is a fast algorithm that approximates the solution of Euler's elastica inpainting problem. Therefore, like elastica inpainting, it performs well for completing smooth regions while maintaining sharp edges but does not complete textures. To overcome this limitation, we propose to combine the constrained elastica inpainting algorithm with texture synthesis. As described in section 5.2, texture synthesis fills in a pixel by copying the center color of a similar patch found through minimizing the weighted sum of squared differences (SSD) with the current patch. The proposed algorithm here uses texture synthesis when a repeating pattern is found and uses elastica energy criteria otherwise to complete the geometry. The decision is determined by a user-defined threshold. For a missing pixel, if the smallest weighted SSD among all candidates is smaller than the threshold, texture synthesis is used for repairing the current pixel. Otherwise, constrained elastica inpainting algorithm is used. The proposed algorithm is outlined as follows:

1. Update the confidence map and the priority map.
2. For the remaining missing pixels, find one with the highest priority.
3. Filling in pixel according to its current neighborhood and searching for similar patches (using weighted SSD). If the current patch is not repeated in existing part, fill in with the color that satisfies the elastica energy criteria.

These steps are iterated until a steady state is reached.

The definition of the neighborhood can be extended to include images forward and backward in time. Often, the exact pixel value for repairing the wire is visible in either the previous or next frame because of motion of the wire, the camera or the background. The combinatorial optimization will quickly choose the replacement pixel because it will satisfy both the Euler's elastica equation and the texture synthesis heuristic.

Using these techniques, we have produced an algorithm that works on both moving and static wire and scratch removal and accelerates the solution of the Euler's elastica inpainting method from hours to less than a minute.

5.5 Numerical Methods

We extrapolate $\overrightarrow{I}$ into the inpainting region D by the proposed texture and geometric inpainting algorithm in section 5.4. The following describes the numerical scheme:

1. Define the confidence map $C : \Omega \to \mathbb{R}$ by

$$C(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \Omega - D \\ 1 - 0.7^0 & \text{otherwise} \end{cases}$$

2. Compute the priority map $P : D \to \mathbb{R}$. The priority value at pixel (i,j) is

$$P(i,j) = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} C(i,j).$$

3. For all missing pixels

4. Find a pixel (i,j) that has not been repaired and with the highest confidence value.

5. Consider the local patch $\omega_{i,j}$ centered at (i,j) with size s. Compute the weighted SSD between $\omega_{i,j}$ and an existing patch $\omega_{k,l}$ within search radius R,

$$d(\omega_{i,j}, \omega_{k,l}) = \sum_{dy=-s}^{s} \sum_{dx=-s}^{s} g(dx,dy) |\overrightarrow{I}_{i+dx,j+dy} - \overrightarrow{I}_{k+dx,l+dy}|^2,$$

where $g(x)$ is a positive, symmetric, and decreasing weight function, such as Gaussian.

6. If

$$\min_{(k-i)^2+(l-j)^2<R^2} d(\omega_{i,j}, \omega_{k,l}) < th,$$

where th is the user-defined threshold, set $\overrightarrow{I}_{i,j} = \overrightarrow{I}_{k^*,l^*}$, where $\omega_{k^*,l^*} =$ argmin $d(\omega_{i,j}, \omega_{k,l})$.

If

$$\min_{(k-i)^2+(l-j)^2<R^2} d(\omega_{i,j}, \omega_{k,l}) \geqslant th,$$

do the following:

Fixing the neighboring pixel colors, for each color candidate $\overrightarrow{v}$, replace $\overrightarrow{I}_{i,j} = \overrightarrow{v}$ and compute the local elastica energy. Denote $u_O = u_{i,j}$, $u_E = u_{i+1,j}$, $u_N = u_{i,j+1}$, $u_W = u_{i-1,j}$, and $u_S = u_{i,j-1}$. The local elastica energy is

$$\sum_{c=R,G,B} \sum_{l=j-1}^{j+1} \sum_{k=i-1}^{i+1} (a + b\kappa_{k,l,c}^2)|\nabla u_{k,l,c}|,$$

where

$$\begin{aligned}
\kappa_{i,j} = & \frac{u_E - u_O}{\sqrt{(u_E - u_O)^2 + \frac{1}{4}(u_{NE} - u_{SE} + u_N - u_S)^2 + \epsilon^2}} \\
& + \frac{u_N - u_O}{\sqrt{(u_N - u_O)^2 + \frac{1}{4}(u_{NE} - u_{NW} + u_E - u_W)^2 + \epsilon^2}} \\
& + \frac{u_W - u_O}{\sqrt{(u_W - u_O)^2 + \frac{1}{4}(u_{NW} - u_{SW} + u_N - u_S)^2 + \epsilon^2}} \\
& + \frac{u_S - u_O}{\sqrt{(u_S - u_O)^2 + \frac{1}{4}(u_{SE} - u_{SW} + u_E - u_W)^2 + \epsilon^2}}
\end{aligned}$$

and

$$|\nabla u_{i,j}| = \frac{1}{\sqrt{2}}\sqrt{(u_E - u_O)^2 + (u_W - u_O)^2 + (u_N - u_O)^2 + (u_S - u_O)^2},$$

with ϵ sufficiently small. After computing the elastica energy for all color candidates, choose the one that results the smallest elastica energy to replace $\overrightarrow{I}_{i,j}$.

7. Set confidence value $C(i,j) = 1 - 0.7^t$, t being current iteration number, update the priority map, and mark current pixel being repaired.

8. Repeat step 4-7 until all missing pixels are filled in.

9. Iterate 8 until no pixel changes its color.

5.6 Experimental Results

Fig. 5.1 shows results of the proposed inpainting algorithm in section 5.4. The original image is in (a), and in (b) the wire is removed by the proposed inpainting algorithm. The sharp edges of the window and the clothes are preserved, as shown in fig. 5.2 (b) and fig. 5.2 (f). The smoothness of the hair is also naturally extended into the occlusion region, in fig. 5.2 (d).

Fig. 5.3 shows another experiment, relating to the removal of objects from the given image in (a). The inpainting region, in (b), is indicated by the user. The results by TV inpainting and the proposed inpainting algorithm are shown in (c) and (d), respectively. Fig.5.4 shows the selected regions of the inpainted results. The proposed method is able to complete the geometry of the clock face (top right in fig. 5.4) and is able to repair corners (bottom right of fig. 5.4) because such structures in the image are repeated. In fig. 5.5, the proposed method produces realistic results by utilizing texture synthesis.

(a) Given image (b) Wire removed

Figure 5.1: Wire removal using the proposed inpainting algorithm.

5.7 Conclusion

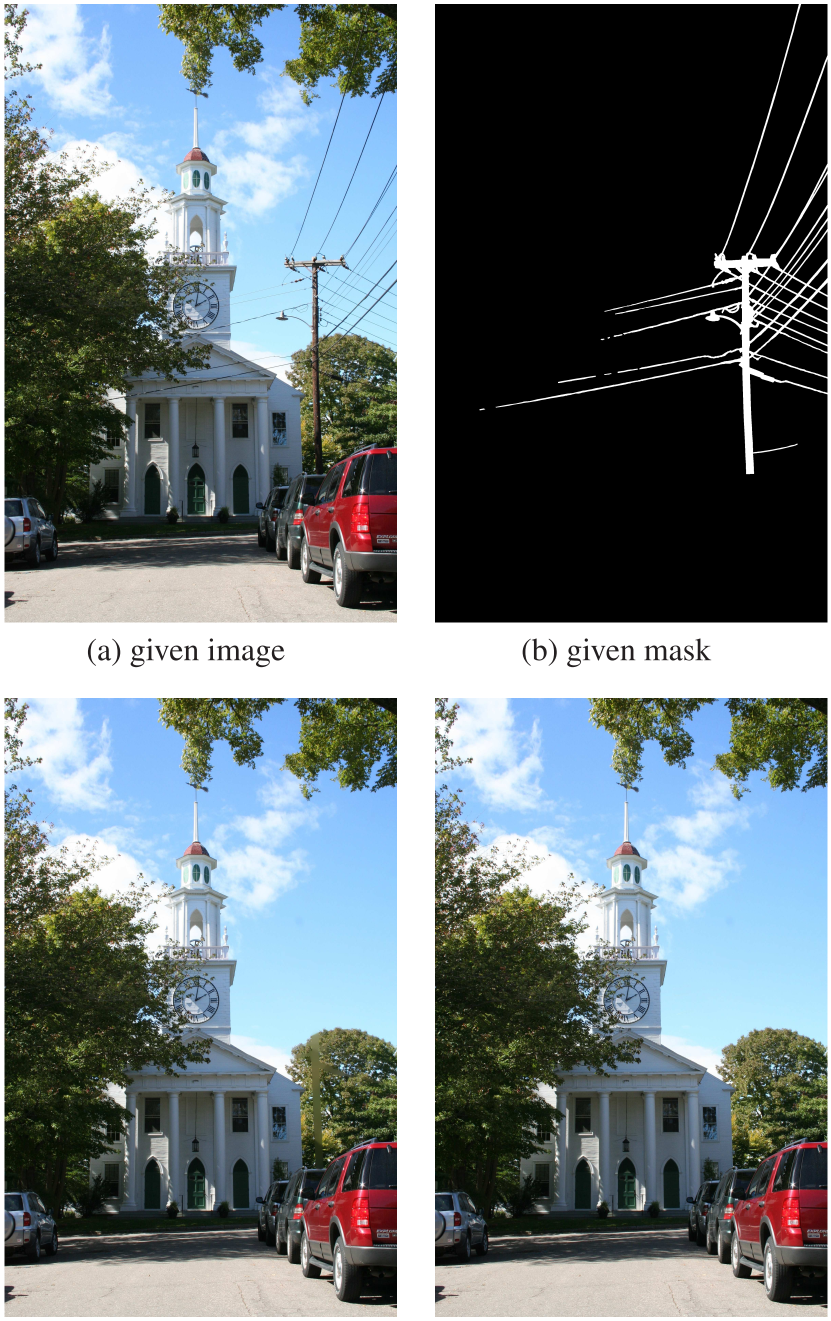
In this chapter, we first proposed a fast algorithm that approximates the solution of Euler's elastica inpainting. Motivated by the texture synthesis techniques, the proposed algorithm turned the original PDE solution into a constrained combinatorial optimization problem. Texture synthesis sometimes can be useful for completing geometric structures, especially when the inpainting region is much larger than the image features. Based on this observation, we combined the proposed algorithm and texture synthesis with a threshold to determine which algorithm should be used for the current damaged pixel. Our experiments show that high quality inpainting results are produced. In the future, we would like to consider a user-interactive algorithm that helps to generate a proper inpainting region without too much manual effort.

(a) Given image (b) Wire removed

(c) Given image (d) Wire removed

(e) Given image (f) Wire removed

Figure 5.2: Selected regions of the wire removal results.



(a) given image (b) given mask

(c) TV inpainting (d) Proposed inpainting

Figure 5.3: Inpainting experiment. The given image (a) has unwanted objects, indicated by the user-defined inpainting region (b). (c) is the result by TV inpainting and (d) is by the proposed inpainting algorithm in section 5.4.

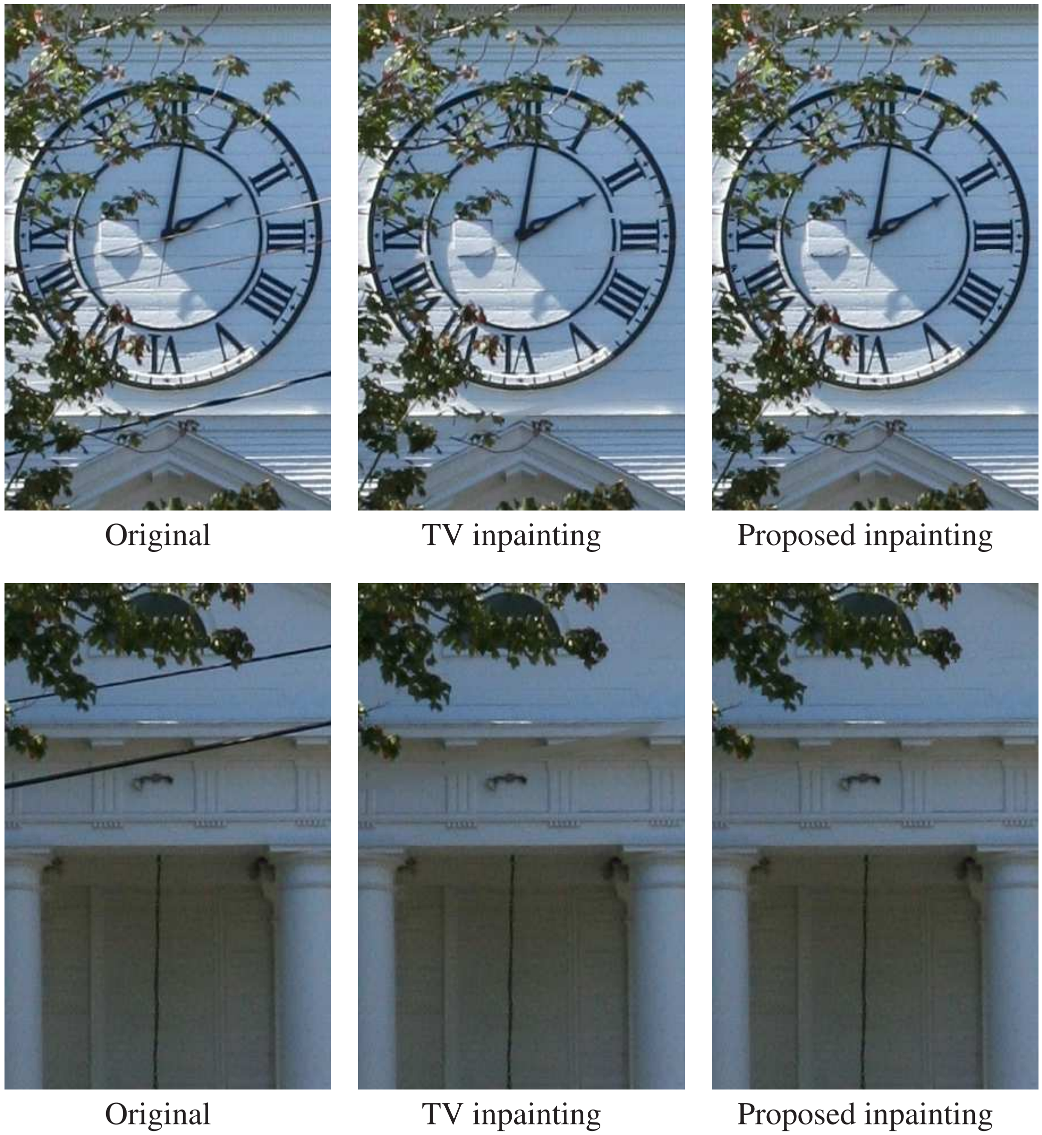


Figure 5.4: Selected regions of the inpainting results. The proposed algorithm is able to complete both geometric structures (clock face), repeating structures (building). On the other hand, TV inpainting is not able to do so when the image features are smaller than the width of the inpainting region.

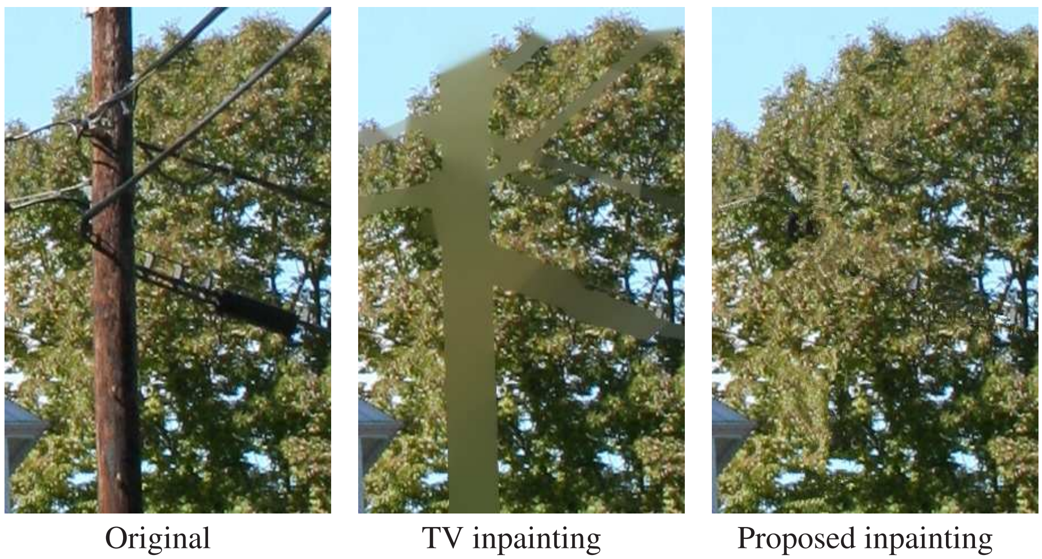


Figure 5.5: Selected region of the inpainting results. Moreover, TV inpainting is not able create oscillating patterns (tree leaves)

CHAPTER 6

Matting

6.1 Introduction

Digital image composition is frequently used by the graphics community to combine objects from different images into a single image. This technique enables filmmakers to film actors and backgrounds separately. For example, the images of actors captured in a controlled studio can be combined with desired backgrounds to produce new images. A crucial step preceding image compositing is to accurately extract the object from its original background; the transparency of the object boundary must be captured. The above step is referred to as image matting, and additionally as in our work, if the background is unknown, it is referred to as natural image matting.

Given a color image $\overrightarrow{I} : \Omega \to [0,1]^3$ containing the object of interest, the matting problem is commonly solved by using the following matting equation:

$$\overrightarrow{I} = \alpha \overrightarrow{F} + (1-\alpha)\overrightarrow{B}.$$

In this equation, $\overrightarrow{F} : \Omega \to [0,1]^3$ is the foreground color corresponding to the object, $\overrightarrow{B} : \Omega \to [0,1]^3$ is the background color, and $\alpha : \Omega \to [0,1]$ is the soft-segmentation (i.e. α-matte) that reflects the transition between the foreground and background. Recovering $(\alpha, \overrightarrow{F}, \overrightarrow{B})$ is an ill-posed problem because there are three unknowns but only one equation. To solve the matting problem, additional a priori assumptions for α, $\overrightarrow{F}$

and $\overrightarrow{B}$ are needed.

One of the early approaches for image matting is blue screen matting [77]. The background is a known constant, yet it is still insufficient to fully constrain the problem. When the background is unknown, a natural matting problem, the object of interest has to be indicated by the user. A common approach is to simplify it by a user-specified *trimap* that partitions the image domain into three regions; as shown in fig.6.1, definite foreground Ω_F, definite background Ω_B, and an unknown transition region D. Thus the problem of recovering (α, $\overrightarrow{F}$, $\overrightarrow{B}$) is restricted only to D. Many previous methods approach it by first imposing statistical priors on ($\overrightarrow{F}$, $\overrightarrow{B}$) [78, 79] and then regularity constraints [80] on α. In Knockout [81], $\overrightarrow{F}$ and $\overrightarrow{B}$ are assumed to be smooth and the estimated α is obtained as a weighted average of $\overrightarrow{F}$ and $\overrightarrow{B}$. Instead of a simple weighted sum as the estimation of a local intensity distribution, a mixture of un-oriented Gaussians has been used in [79] and principal component analysis has been used in [82]. In [78], Chuang et al. proposed a Bayesian matting algorithm where a mixture of oriented Gaussians is employed to estimate the local distribution and $\overrightarrow{F}$, $\overrightarrow{B}$ and α are estimated by a maximum a posterior in a Bayesian framework. The Poisson matting algorithm was proposed in [80], where the alpha matte is obtained from its gradient field by solving a Poisson equation using the boundary information provided by a trimap. Robust matting [83] samples foreground and background colors for each unknown pixels and analyzes the confidence of the samples. The alpha estimate is then optimized by a Random Walk using the the confidence values of the samples and enforcing the smoothness of the matte. These methods generally produce good results, but have a major drawback. When the foreground and background has sharp discontinuities, the performance is unreliable. The nearby pixel regions and do not take into account the inherent geometry. For example, in Fig.6.2, (a) is an image occluded by an object (gray region). (b) shows the desired background estimation.

(c) is the result using nearest neighbor interpolation, variants of which are used in current matting algorithms such as Poisson matting [80]. We see from (c) that such methods are not able to achieve correct inpainting results. Hence, the extracted mattes may be erroneous even for images with simple geometric structures. This drawback is especially severe when the foreground color and background color are similar.

We propose to solve this issue by utilizing variational inpainting algorithms to solve the matting problem for non-textured images [84]. Since the geometric structures are accurately extrapolated into the unknown regions, the extracted matte solved by the matting equation is reliable. However, this method cannot be used for images with oscillating patterns, such as textures. Therefore, we utilize the proposed inpainting algorithm in Chapter 5, which is able to correctly connect edges of geometric structures into the unknown region and also complete repeating patterns. The proposed matting algorithm consists of three steps, alternating the solutions of F, B, and α in the unknown region. The first step is to update the unknown region according to the obtained α estimation. In the second step, the foreground and the background are extrapolated by the proposed inpainting algorithm. In the third step, α is solved by the proposed variational model for α that consists of a fidelity term, using the matting equation, and a regularization term, searching for α in the Sobolev space $H^1(\Omega)$.

We here mention some other natural matting approaches that are out of the scope of this chapter. These methods are concerned with the problem of the trimap simplification approaches, in which solutions are considerably dependent on the user specified trimap. Generating an accurate trimap could be time-consuming and intractable. Some of the recent works combine image segmentation and matting. One approach is based on a few user-specified strokes rather than on trimap using Belief Propagation techniques [85] and Markov Random Field [86] for modeling unknown regions [86, 87, 85]. In [88], the matte is solved by a variational PDE based model for non-

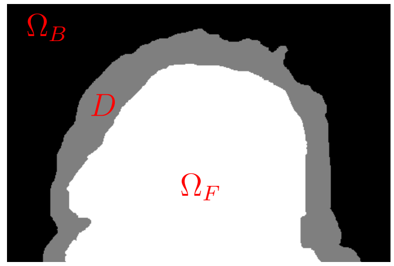
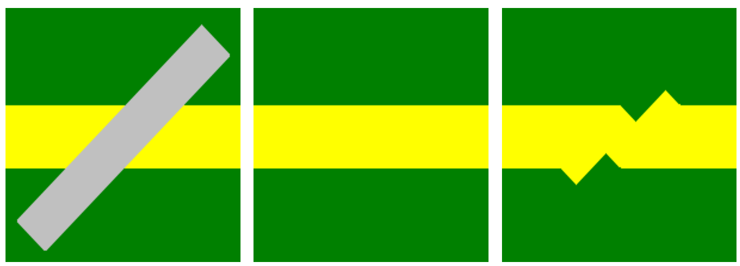


Figure 6.1: A user define a trimap that partitions the image domain Ω into three regions: definite foreground Ω_F, definite background Ω_B, and unknown region$D$

texture images. This model is robust with respect to the initial trimap, because of the additional priors on α.



(a) damaged (b) desired (c) nearest interp.

Figure 6.2: Failure example of interpolating the missing (gray) region of an image by using the nearest known pixel values.

The organization of the remaining chapter is the following. We next describe related works, Poisson matting [80], that takes a variational approach to solve α within the matting problem and robust matting [83]. Section 6.2 presents the proposed matting algorithm that employs TV inpainting for gray-scale images. We also propose a variational model for solve α. Section **??** shows the proposed matting algorithm that utilizes geometric and texture inpainting for color images. In section 6.4, we show several results of the proposed algorithm and the related works. Finally, we conclude this chapter in section 6.5.

6.1.1 Related work: Poisson matting

Poisson matting [80] solves the matting problem by a three-step iterative scheme: updating the unknown region, extrapolating F and B, and solving for α. The alpha matte is solved by a variational PDE-based model. This model is derived from the matting equation with the assumption that the foreground and background of a given image are smooth. Taking the gradient of the matting equation and approximating ∇F and ∇B by zero, one gets

$$\nabla \alpha \approx \frac{1}{F-B} \nabla I.$$

According to this equation, Poisson matting is to solve the following energy minimization problem:

$$\min_{\alpha} \int ||\nabla \alpha(x) - \frac{1}{F(x)-B(x)} \nabla I(x)||^2 dx, \tag{6.1}$$

with the boundary conditions:

$$\alpha|_{\Omega_F} = 1 \text{ and } \alpha|_{\Omega_B} = 0. \tag{6.2}$$

The Euler-Lagrange equation of energy in (6.1) is:

$$\triangle \alpha = \text{div}(\frac{\nabla I}{F-B}). \tag{6.3}$$

Poisson matting is summarized in the following iterative scheme:

1. **Trimap refinement:** A prior segmentation step refines the user defined trimap to conform to the actual α-transition region. Update $\Omega_F = \{x \in \Omega | \alpha(x) > 0.95, I(x) \approx F(x)\}$ and $\Omega_B = \{x \in \Omega | \alpha(x) < 0.05, I(x) \approx B(x)\}$.

2. **Extrapolating F and B:** The foreground and background intensities, F and B are extrapolated into the transition region by the nearest foreground value and the nearest background value, respectively. Then, the constructed $F - B$ is smoothed by a Gaussian filter.

3. **Solving for α:** α is solved in the transition region by (6.3) with boundary conditions (6.2).

These steps are repeated until convergence. For color images, the implementation is on a single channel to obtain the alpha matte.

6.1.2 Related work: Robust matting

Robust matting [83] is based on an optimized color sampling scheme and is more robust for natural images. The algorithm first samples a large number of foreground and background samples to estimate true foreground and background colors for each unknown pixel. In particular, the foreground and background samples are spread along the known boundaries, instead to using nearby pixels, to fully capture the variation of the foreground and background colors. For each pair of foreground and background samples, F and B, the estimated alpha value is

$$\alpha = \frac{(I - B)(F - B)}{||F - B||^2}.$$

The confidence of each pair is analyzed, and only high confidence samples are chosen to obtain an initial alpha estimate. The initial alpha estimate is then further improved by a graph-labeling problem which is minimized by a Random Walk. The graph-labeling problem enforces the smoothness of the matte in addition to the data constraint for each pixel. The balance between the data term and smoothness term is controlled by the confidence value; if the confidence value is high, the alpha value depends on the matting equation from the selected samples more than the smoothness constraint. The constructed graph-labeling problem is solved by a Random Walk to minimize the total graph energy.

Even though robust matting is able to produce mattes quite accurately for natural images, we observed that the foreground and background sampling method is not accurate near sharp gradients in the foreground, which leads to inaccurate foreground extraction. The reason is also that the confidence value does not take into account the geometry of the foreground and/or background.

6.2 Matting through Variational Inpainting

Let $I : \Omega \to [0, 1]$ be the given image. A trimap is provided by the user, indicating the definite background region Ω_B, the definite foreground region Ω_F and the unknown region D, see figure 6.1. The proposed matting algorithm is the following iterative scheme, $n = 1, 2, \ldots$.

1. **Trimap refinement:**

$$\Omega_F^{(n)} = \{x \in \Omega : \alpha^{(n-1)}(x) > 0.95\}$$

$$\Omega_B^{(n)} = \{x \in \Omega : \alpha^{(n-1)}(x) < 0.05\}$$

2. **Extrapolating F and B:**

$$B^{(n)} = \arg\min_B E_{tv}[B], \text{ with } B|_{\Omega_B^{(n)}} = I|_{\Omega_B^{(n)}}$$

$$F^{(n)} = \arg\min_F E_{tv}[F], \text{ with } F|_{\Omega_F^{(n)}} = I|_{\Omega_F^{(n)}}$$

3. **Solving for α:**

$$\alpha^{(n)} = \arg\min_{\alpha} \int (\alpha F^{(n)} + (1-\alpha)B^{(n)} - I)^2 + \frac{\lambda}{2}|\nabla\alpha|^2,$$

with constraint $\alpha|_{\Omega_F^{(n)}} = 1$ and $\alpha|_{\Omega_B^{(n)}} = 0$

Steps $1-3$ are repeated until α converges, i.e. $d(\alpha^{(n)}, \alpha^{(n+1)}) < \epsilon$, where d, for example, can be the l_1-norm and ϵ is small.

The first subsection describes the second step that extrapolates background $B : \Omega \to [0,1]$ and foreground $F : \Omega \to [0,1]$ through the total variation. The background is obtained by inpainting the data on the definite background Ω_B into the unknown region D. Similarly, the foreground is obtained by inpainting the data on the definite foreground Ω_F into D. The second subsection describes the variational model in step 3 for extracting the matte.

6.2.1 Total variation inpainting

As explained in Chapter 5, TV inpainting is a PDE-based variational model, adapted from the ROF denoising model [52]. It is based on the observation that edges play an important role in the geometry of an image. The TV inpainting interpolates images across the missing regions, while preserving sharp edges. We propose to utilize this technique to extrapolate the background and foreground for the matting problem. The extrapolated background through the TV inpainting is obtained by the following energy minimization:

$$\min_{B \in BV(D \cup \Omega_B)} E_{tv}[B] = \int_{D \cup \Omega_B} |\nabla B|, \tag{6.4}$$

with constraint

$$B\mid_{\Omega_B}=I\mid_{\Omega_B}. \tag{6.5}$$

The gradient descent of the Euler-Lagrange equation for the energy minimization problem (6.4) is

$$\frac{\partial B}{\partial t}=1_{D\cup\Omega_B}\nabla\cdot\left(\frac{\nabla B}{|\nabla B|}\right), \tag{6.6}$$

with constraint (6.5). The boundary condition along the boundary between D and Ω_F is

$$\frac{\partial B}{\partial\overrightarrow{\nu}}=0.$$

The formulation for estimating the foreground can be derived similarly.

6.2.2 Matte extracting

The α matte according to the estimated foreground and background is extracted by minimizing the following proposed variational problem:

$$\min_{\alpha}\int(\alpha F+(1-\alpha)B-I)^2+\frac{\lambda}{2}|\nabla\alpha|^2dx\ , \tag{6.7}$$

with constraints

$$\alpha\mid_{\Omega_F}=1\text{ and }\alpha\mid_{\Omega_B}=0.$$

This imposes smoothness of α while the composite image is close to the given image.

6.2.3 Convergence proof of the iterative scheme

Assumptions: The ground truth image I is a linear combination of F and B, both in the BV space.

In step 2 of our iterative scheme (see Sec.6.2), the existence of a noise-free TV inpainting is proved in [6]. In step 3, existence and uniqueness can be proved in a standard argument (by coercivity, lower semi-continuity, and strict convexity). In step 1, $\{\Omega_{F,n}\}$ and $\{\Omega_{B,n}\}$ are constructed according to $\alpha \in H^1$, so they are measurable. Observe also that by the construction of the iterative scheme, the sequences $\{\Omega_{F,n}\}$ and $\{\Omega_{B,n}\}$ are increasing and bounded by $\Omega_F \cup D$ and $\Omega_B \cup D$, respectively. Therefore,

$$1_{\Omega_{F,n}} \leq 1_{\Omega_{F,n+1}},$$

$$1_{\Omega_{F,n}} \to 1_{\cup \Omega_{F,n}} \text{ a.e. pointwise,}$$

$$1_{\Omega_{B,n}} \leq 1_{\Omega_{B,n+1}},$$

$$1_{\Omega_{B,n}} \to 1_{\cup \Omega_{B,n}} \text{ a.e. pointwise.}$$

By the Monotone Convergence Theorem,

$$1_{\Omega_{F,n}} \to 1_{\cup \Omega_{F,n}} \text{ and } 1_{\Omega_{B,n}} \to 1_{\cup_n \Omega_{B,n}} \text{ in } L^1.$$

Claim 1: $\{F_n\}$ and $\{B_n\}$ are Cauchy sequences in $L^1(D)$.

Proof: We will show Claim 1 for the sequence $\{F_n\}$. The proof for $\{B_n\}$ is similar. First, note that

$$F_n = \text{argmin} \int 1_{D \setminus \Omega_{F,n}} |\nabla F|, \text{ with constraint } F|_{\Omega_{F,n}} = I|_{\Omega_{F,n}}$$

$$F_{n+1} = \text{argmin} \int 1_{D \setminus \Omega_{F,n+1}} |\nabla F|, \text{ with constraint } F|_{\Omega_{F,n+1}} = I|_{\Omega_{F,n+1}}$$

Let $E_n[F] = \int_D |\nabla F|$. We will show that given $\epsilon > 0$,

$$E_n[F_n] \leq E_{n+1}[F_{n+1}] \leq E_{n+1}[F_n] \leq E_n[F_n] + \epsilon \text{ , for n large enough. } (*)$$

The first two inequalities of $(*)$ are straightforward. Write

$$E_{n+1}[F_n] = \int 1_{D \setminus \Omega_{F,n+1}} |\nabla F_n| + 1_{\Omega_{F,n+1} \setminus \Omega_{F,n}} |\nabla I| + 1_{D \cap \Omega_{F,n}} |\nabla I|$$

and

$$E_n[F_n] = \int 1_{D \setminus \Omega_{F,n}} |\nabla F_n| + 1_{\Omega_{F,n+1} \setminus \Omega_{F,n}} |\nabla F_n| + 1_{D \cap \Omega_{F,n}} |\nabla I|.$$

Since $\exists N$, such that $|\Omega_{F,n+1} - \Omega_{F,n}| < \epsilon$, $\forall n \geq N$, we have

$$E_{n+1}[F_n] - E_n[F_n] = \int_{\Omega_{F,n+1} \setminus \Omega_{F,n}} |\nabla I| - |\nabla F_n| \leq \epsilon,$$

up to a scalar constant, depending only on I. This proves the third inequality of $(*)$.

Immediately, $(*)$ gives $0 \leq E_{n+1}[F_{n+1}] - E_n[F_n] \leq \epsilon$, i.e. $0 \leq \int_D |\nabla F_{n+1}| - |\nabla F_n| \leq \epsilon$. With our construction that F_n is the initial state of F_{n+1} in the minimization procedure,

$$\int_D |\nabla F_{n+1} - \nabla F_n| \rightarrow 0.$$

Note that we take n large enough to pass critical situations*, if necessary.

Since $F_{n+1} = F_n$ on ∂D, by Poincaré inequality,

$$\int_D |F_{n+1} - F_n| < C \int_D |\nabla F_{n+1} - \nabla F_n|, \text{ for some constant C.}$$

Therefore, $\{\nabla F_n\}$ is a Cauchy sequence in $L^1(D)$. $\square$

Claim 2: $\{\alpha_n\}$ converges in $H^1(D)$.

proof: By Claim 1, since $L^1(D)$ is complete, F_n and B_n converges, say to $\hat{F}$ and $\hat{B}$, respectively. Let

$$E_n[\alpha] = \int_D (\alpha F_n + (1-\alpha)B_n - I)^2 + \frac{\lambda}{2}|\nabla\alpha|^2 \text{ and}$$

$$\hat{E}[\alpha] = \int_D (\alpha \hat{F} + (1-\alpha)\hat{B} - I)^2 + \frac{\lambda}{2}|\nabla\alpha|^2.$$

Then,

$$|E_n[\alpha]-\hat{E}[\alpha]| = \int |\alpha(F^n-\hat{F})+(1-\alpha)(B^n-\hat{B})||\alpha(F^n+\hat{F})+(1-\alpha)(B^n+\hat{B})-I|.$$

The first factor in the integral is less than $|F^n - \hat{F}| + |B^n - \hat{B}|$ and the second factor is bounded. Thus, $E_n[\alpha] \to \hat{E}[\alpha]$.

Recall that

$$\alpha_n = \text{argmin } E_n[\alpha], \text{ with constraints } \alpha|_{\Omega_{F,n}} = 1 \text{ and } \alpha|_{\Omega_{B,n}} = 0,$$

and let

$$\hat{\alpha} = \text{argmin } \hat{E}[\alpha], \text{ with constraints } \alpha|_{\Omega_{\hat{F}}} = 1 \text{ and } \alpha|_{\Omega_{\hat{B}}} = 0.$$

We will show that $\alpha_n \to \hat{\alpha}$ in $L^2(D)$ (stability of minimizers).

Since

$$\text{lim sup } E_n[\alpha_n] \leq \text{ lim sup } E_n[\hat{\alpha}] = \hat{E}[\hat{\alpha}],$$

$$\{\int_D |\nabla \alpha_n|^2\} \text{ is bounded.}$$

Moreover, since $0 \leq \alpha_n \leq 1$ and D is bounded, $\{\alpha_n\}$ is bounded in $L^2(D)$. So $\{\alpha_n\}$ is bounded in $H^1(D)$.

Suppose on the contrary that α_n does not converge to $\hat{\alpha}$ in $L^2(D)$. By the Rellich-Kondrachov Compactness Theorem, there is a subsequence $\{\alpha_{n_k}\}$ that converges in $L^2(D)$ to a function $\bar{\alpha} \in L^2(D)$. Moreover, $\{\alpha_{n_k}\}$ also contains a subsequence (which we continue to denote by $\{\alpha_{n_k}\}$) that converges weakly to some $\alpha' \in H^1(D)$, $\alpha_{n_k} \rightharpoonup \alpha'$. (Since every bounded sequence in a Hilbert space contains a weakly convergent subsequence.) But $\alpha_{n_k} \to \bar{\alpha}$ in $L^2(D)$ implies that we must have $\bar{\alpha} = \alpha'$. So we have $\int_D |\nabla \bar{\alpha}|^2 \leq$ lim inf $\int_D |\nabla \alpha_{n_k}|^2$.

Therefore,

$$\begin{aligned}
\hat{E}[\bar{\alpha}] &\leq \text{ lim inf } \hat{E}[\alpha_{n_k}] \\
&= \lim(\hat{E}[\alpha_{n_k}] - E_{n_k}[\alpha_{n_k}]) + \text{ lim inf } E_{n_k}[\alpha_{n_k}] \\
&\leq 0 + \text{ lim inf } E_{n_k}[\hat{\alpha}] \\
&= \hat{E}[\hat{\alpha}].
\end{aligned}$$

The last two inequalities are by $E_{n_k}[\alpha_{n_k}] \leq E_{n_k}[\hat{\alpha}]$ and $E_n[\alpha] \to \hat{E}[\alpha]$.

This contradicts to the uniqueness solution of argmin $\hat{E}[\alpha]$. Therefore, $||\alpha_n - \hat{\alpha}||_{L^2(D)} \to 0$.

Finally, we will prove that $||\nabla \alpha_{n+1} - \nabla \hat{\alpha}||_{L^2(D)} \to 0$

Since $\alpha_n =$ argmin $E_n[\alpha]$ and $\hat{\alpha} =$ argmin $\hat{E}[\alpha]$, α_n and $\hat{\alpha}$ satisfy the Euler-Lagrange equations associated with $E_n[\alpha]$ and $\hat{E}[\alpha]$, respectively:

$$\triangle \alpha_n = \frac{1}{\lambda}(\alpha_n F_n + (1-\alpha_n)B_n - I)(F_n - B_n)$$

and

$$\triangle \hat{\alpha} = \frac{1}{\lambda}(\hat{\alpha}\hat{F} + (1-\hat{\alpha})\hat{B} - I)(\hat{F} - \hat{B}).$$

Then,

$$\begin{aligned}
&\int |\nabla\hat{\alpha} - \nabla\alpha_n|^2 \\
&\quad = \int (\nabla\hat{\alpha} - \nabla\alpha_n)\cdot(\nabla\hat{\alpha} - \nabla\alpha_n) \\
&\quad = -\int \triangle(\hat{\alpha} - \alpha_n)(\hat{\alpha} - \alpha_n) \\
&\quad = -\frac{1}{\lambda}\int \{(\alpha_n F_n + (1-\alpha_n)B_n - I)(F_n - B_n) - (\hat{\alpha}\hat{F} + (1-\hat{\alpha})\hat{B} - I)(\hat{F} - \hat{B})\}(\hat{\alpha} - \alpha_n) \\
&\quad \le \frac{1}{\lambda}\int M|\hat{\alpha} - \alpha_n|, \text{ for some constant } M \\
&\quad \le M||\hat{\alpha} - \alpha_n||_{L^2(D)}|D|.
\end{aligned}$$

Thus, $||\hat{\alpha} - \alpha_n||_{H^1(D)} \to 0$. $\square$

Therefore, our proposed iterative scheme converges.

$*$ A *critical situation* of a TV inpainting scenario is when there are more than one global minimizers. The occurrence of this situation depends on the inpainting domain. Given a BV image, there are at most finitely many critical situations over a set of decreasing inpainting regions.

6.3 Matting through Texture and Geometric Inpaintping

In this section, we propose to use the texture and geometric inpainting algorithm described in the previous chapter within the matting algorithm. The inpainting algorithm is used to extrapolate the background color $\overrightarrow{B} : \Omega \rightarrow [0,1]^3$ and the foreground color $\overrightarrow{F} : \Omega \rightarrow [0,1]^3$ into the unknown region D. The proposed matting algorithm is the following iterative scheme, $n = 1, 2, \ldots$.

1. **Trimap refinement:**

$$\Omega_F^{(n)} = \{x \in \Omega : \alpha^{(n-1)}(x) > 0.95\}$$

$$\Omega_B^{(n)} = \{x \in \Omega : \alpha^{(n-1)}(x) < 0.05\}$$

2. **Extrapolating F and B:**

   Use texture and geometric inpainting described in section 5.4 to obtain $\overrightarrow{B}^{(n)}$ and $\overrightarrow{F}^{(n)}$ with boundary constraints

$$\overrightarrow{B}|_{\Omega_B^{(n)}} = \overrightarrow{I}|_{\Omega_B^{(n)}}$$

   and

$$\overrightarrow{F}|_{\Omega_F^{(n)}} = \overrightarrow{I}|_{\Omega_F^{(n)}},$$

   respectively.

3. **Solving for α:**

$$\min_{\alpha} \int_D |\alpha\overrightarrow{F} + (1-\alpha)\overrightarrow{B} - \overrightarrow{I}|^2 + \lambda|\nabla\alpha|^2 dx \; , \tag{6.8}$$

with constraints

$$\alpha \mid_{\Omega_F}= 1 \text{ and } \alpha \mid_{\Omega_B}= 0,$$

where λ is a parameter and $|\cdot|$ denotes the Euclidean distance.

Initially, $\alpha^{(0)}$ is given by the user; $1, 0.5, 0$ represent definite foreground, unknown region, and definite background, respectively. The matting algorithm repeats step $1-3$ until α converges, i.e. $d(\alpha^{(n)}, \alpha^{(n+1)}) < \epsilon$, where d for example can be the l_1-norm and ϵ is small.

In step 2, $\overrightarrow{B}$ is extrapolated into the unknown region D by the proposed texture and geometric inpainting algorithm in section 5.4. The numerical implementation follows section 5.5 directly, except for the following minor adaption:

The confidence map $C : \Omega \to \mathbb{R}$ is defined by

$$C(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \Omega_{B^{(n)}} \\ 0 & \text{if } (i,j) \in \Omega_{F^{(n)}} \\ 1 - 0.7^0 & \text{otherwise} \end{cases}$$

The confidence map in extrapolating $\overrightarrow{F}$ into the unknown region D can be defined simply by replacing $\Omega_B^{(n)}$ by $\Omega_F^{(n)}$.

In step 3, we propose a variational model (6.8) to extract the matte, using the matting equation and current $\overrightarrow{F}$ and $\overrightarrow{B}$. The constraints force α to equal 1 in the definite foreground region and equal 0 in the definite background region. The first term in the energy seeks an α that adheres to the matting equation. This term alone is not enough to extract α matte. Even when $\overrightarrow{F}$ and $\overrightarrow{B}$ are correct, the solution α of the matting equation is not unique. This occurs when $\overrightarrow{F}$ and $\overrightarrow{B}$ at some pixels have the same colors. Numerically, the solutions are unstable and thus a regularization must be imposed. The second term enforces smoothness of α, according to parameter

λ. We look for a solution in the Sobolev space, $\alpha \in H^1(\Omega)$. This proposed energy minimization problem is strictly convex and there exists a unique solution for fixed $\overrightarrow{F}$ and $\overrightarrow{B}$. The solution, α, is evolved by the gradient flow of the energy in (6.8) until convergence, shown in the following

$$\frac{\partial \alpha}{\partial t} = -\Big[\alpha \overrightarrow{F} + (1-\alpha)\overrightarrow{B} - I\Big](\overrightarrow{F} - \overrightarrow{B}) + \lambda \bigtriangleup \alpha. \tag{6.9}$$

Numerically, the solution of α matte is obtained by the steepest descent of Euler-Lagrange equation shown in (6.9), which is discretized in the following:

$$\frac{\alpha^{n+1} - \alpha^n}{\delta t} = \sum_{c=R,G,B} \Big\{\Big[\alpha^n F_c + (1-\alpha^n)B_c - I_c\Big](F_c - B_c)\Big\} - \lambda \triangle \alpha^n,$$

where

$$\triangle \alpha = -4\alpha_{i,j} + \alpha_{i+1,j} + \alpha_{i-1,j} + \alpha_{i,j+1} + \alpha_{i,j-1}$$

is the usual five point discrete Laplacian.

6.4 Experimental Results

In this section, we show experimental results of the proposed matting algorithm on various natural images and compare them with experimental results of Poisson matting and robust matting.

Fig.6.3 shows an experimental result of the proposed matting algorithm. The object of the given image (b) occludes a geometric structure in the background. The trimap given by the user is shown in (a). Column (d) is the results by Poisson matting, column

(e) is the results by robust matting, and column (f) is the results by the proposed algorithm. The top row shows the extracted mattes by each method, and Poisson matting is not accurate near sharp gradients in the background. The extracted mattes by robust matting and the proposed algorithm are accurate. The second row shows the extracted foregrounds by each method. The third row shows a selected region of the extracted foregrounds. Poisson matting, shown in bottom (d), is erroneous near the boundary of the bear and geometric structure. The reason is that the nearest neighbor interpolation fails to accurately recover the underlying geometry of the background.

In the matting experiment in Fig.6.4, the object of the given image (b) occludes a checkerboard, as the background. The trimap given by the user is shown in (a). The first and second rows show the extracted mattes and extracted foreground by Poisson matting and the proposed matting algorithm, respectively. The proposed matting algorithm outperforms Poisson matting and robust matting because the former is able to complete the texture structure in the unknown region. This can be seen in the bottom row, in which we illustrate the selected local regions of Poisson matting (d), robust matting (e), and the proposed matting (f).

Fig. 6.5(b) has a highly oscillating background and the object's hair color is similar to part of the background. The trimap is shown in (a). Poisson matting, in column (d), is erroneous due to the smoothness assumption on the foreground and background. Robust matting, in column (e), is able to extract an accurate alpha matte but the extracted foreground is not as good as that of the proposed matting, in column (f). In fig. 6.6, the extracted foreground by robust matting is erroneous near color transitions in the foreground.

In the matting experiment in Fig. 6.7, the proposed matting algorithm is able to accurately extract the object and outperforms Poisson matting and robust matting. Fig. 6.8 is an application of matting. (a) and (b) are two images with similar backgrounds.

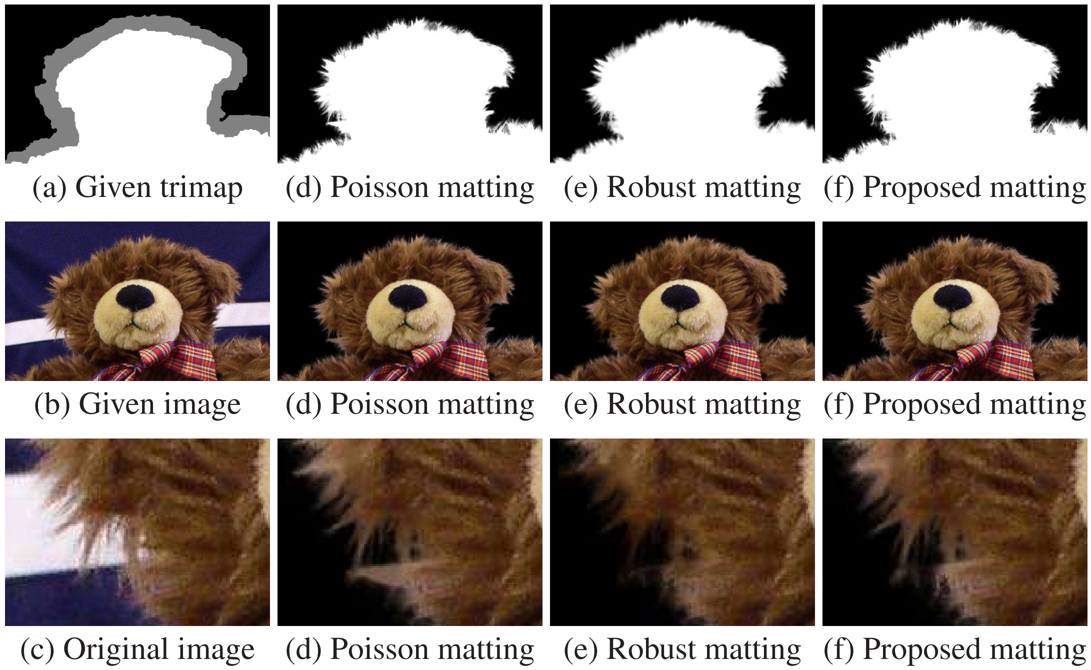


Figure 6.3: Matting experiment. The object of the given image (a) occludes the geometric structure in the background. (b) is the given trimap. (c) and (d) are the extracted mattes by Poisson matting and the proposed matting algorithm, respectively. (e) and (f) are the extracted foregrounds by Poisson matting and the proposed matting algorithm, respectively. The proposed algorithm is able to accurately extract the foreground near sharp gradients in the background because it utilizes Euler's elastica inpainting in the extrapolating step.

The object of (a) is extracted by the proposed matting algorithm (c) and then is used to composite with image (b). The final result (d) is realistic.

The above results demonstrate that when the background has either geometric or texture structures occluded by the foreground, an inpainting algorithm that works for both geometry and texture regions of an image, are able to recover the background correctly. Consequently, the extracted matte is more accurate. From our experiment, the extracted mattes from the iterative scheme do not change significantly after about 3 iterations. The numerical error converges in about 30 iterations.

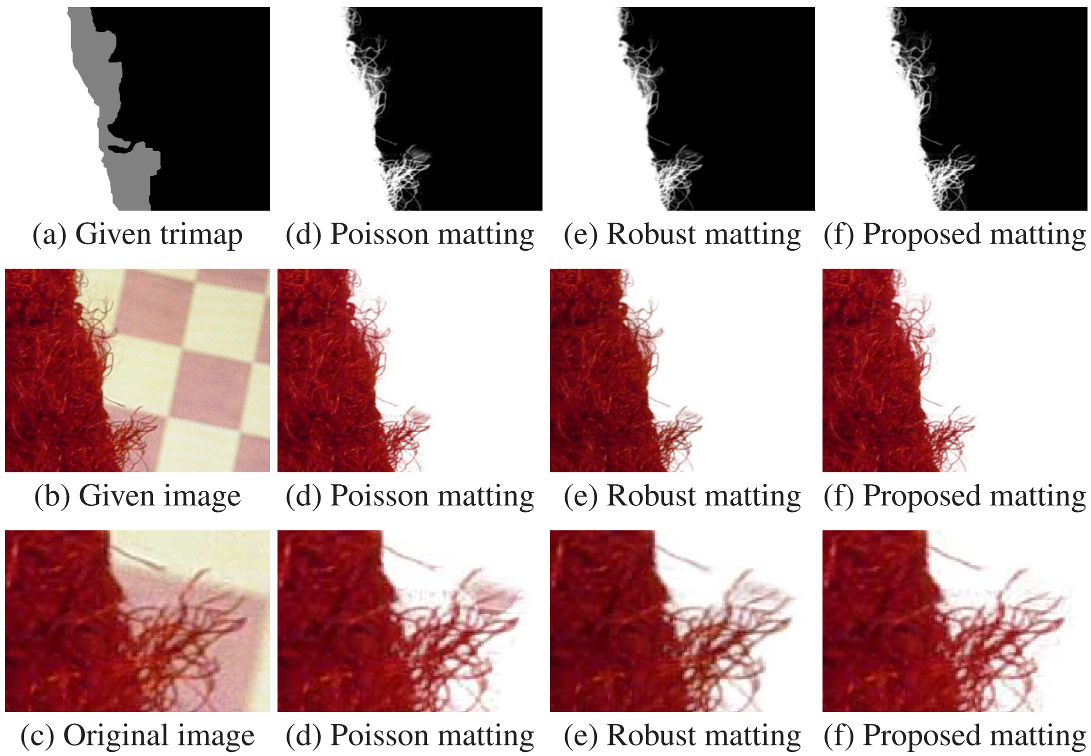
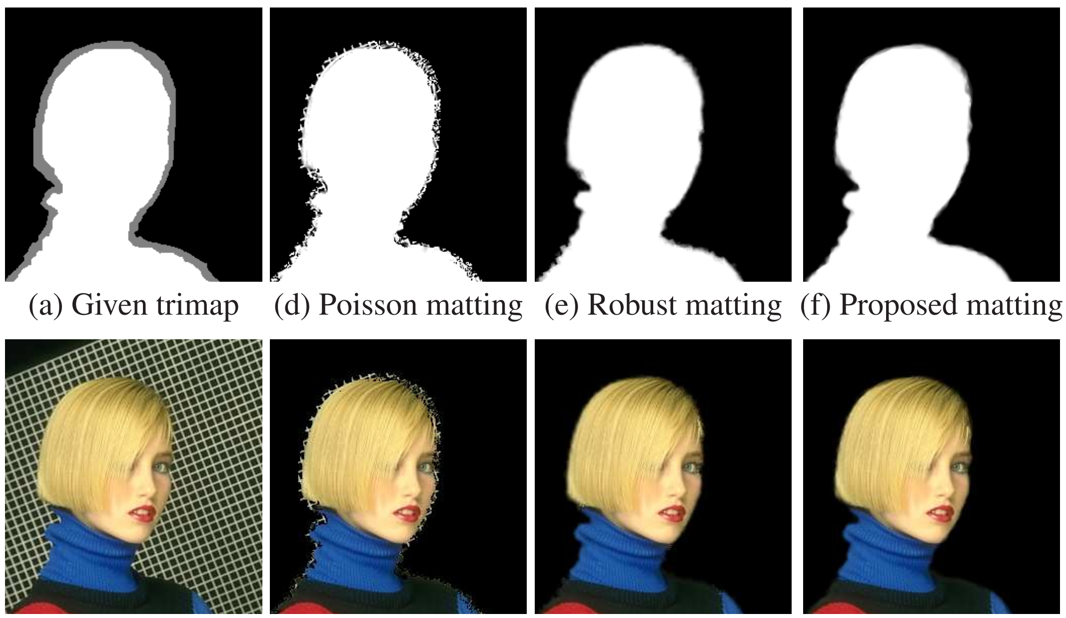


Figure 6.4: The proposed matting algorithm produces an accurate matte of the object.

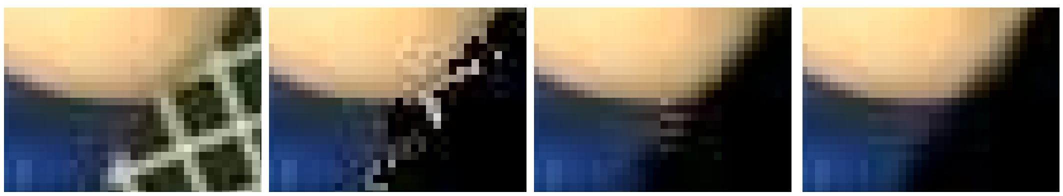


(a) Given trimap (d) Poisson matting (e) Robust matting (f) Proposed matting

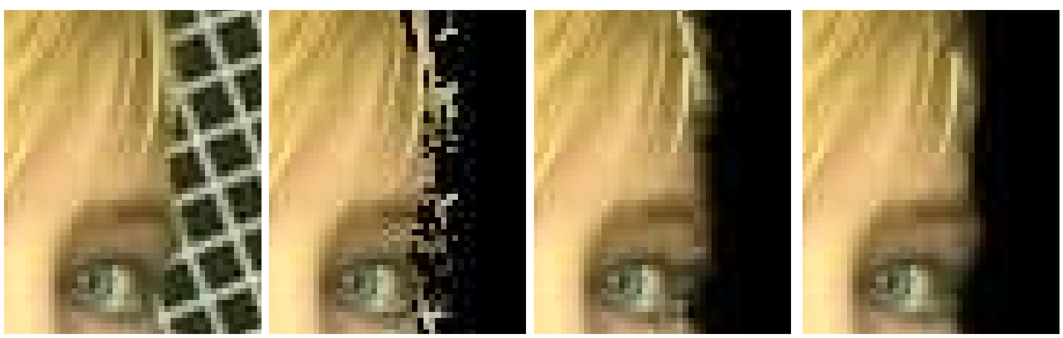
(b) Given image (d) Poisson matting (e) Robust matting (f) Proposed matting

Figure 6.5: In this matting experiment, the background of the given image (a) has sharp gradients. Moreover, the foreground and background have similar colors. The extracted matte by Poisson matting (e) is erroneous because the nearest neighbor interpolating is not correct and also due to the smoothness assumptions on the foreground and background.

(c) Original image (d) Poisson matting (e) Robust matting (f) Proposed matting



(c) Original image (d) Poisson matting (e) Robust matting (f) Proposed matting



(c) Original image (d) Poisson matting (e) Robust matting (f) Proposed matting

Figure 6.6: Robust matting (e) gives a reasonable solution because there is no smoothness assumptions on the background and/or foreground. However, the extracted foreground is not as good as that of the proposed matting algorithm (f), which additionally correctly inpaints the background and foreground into the unknown region.

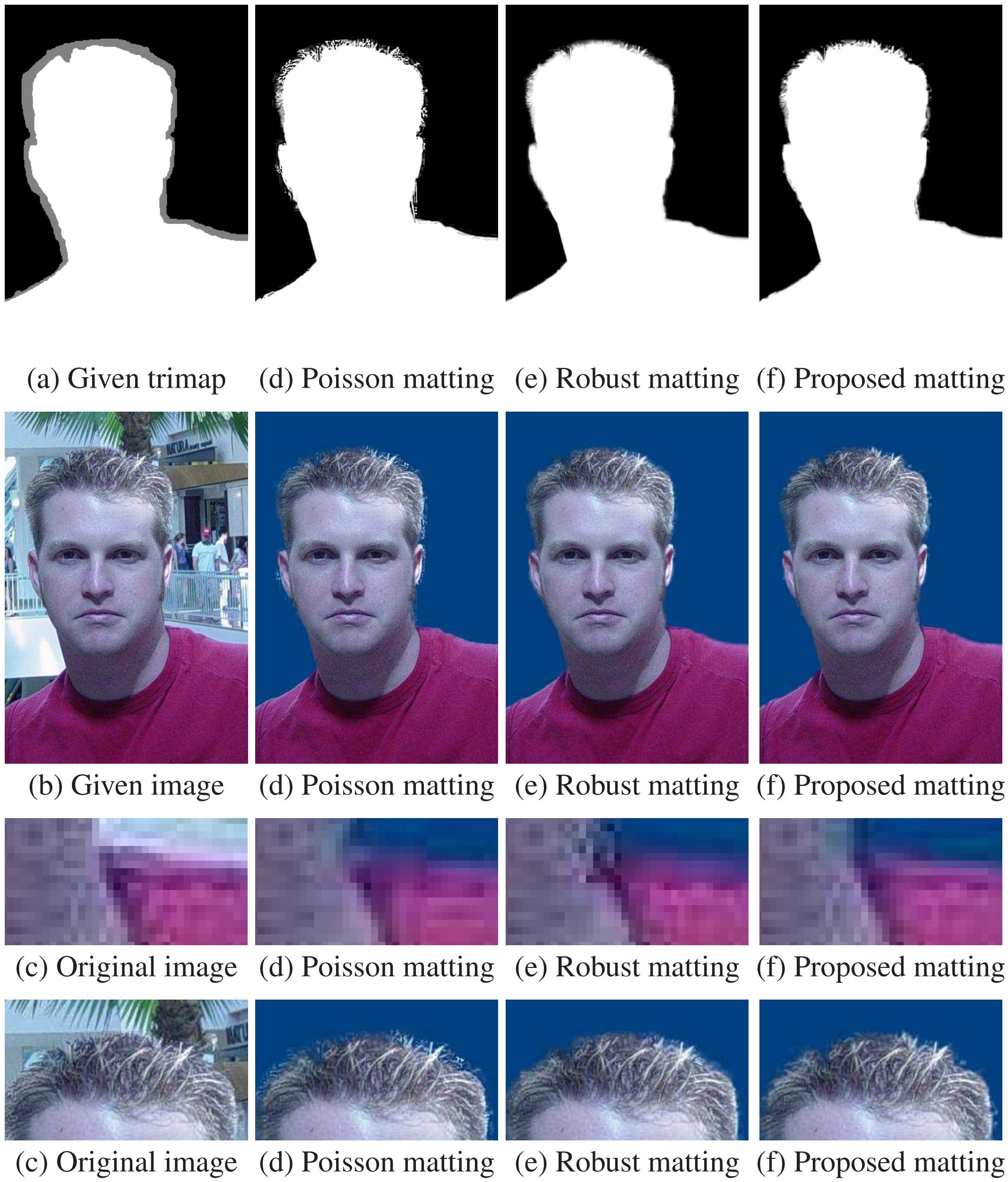
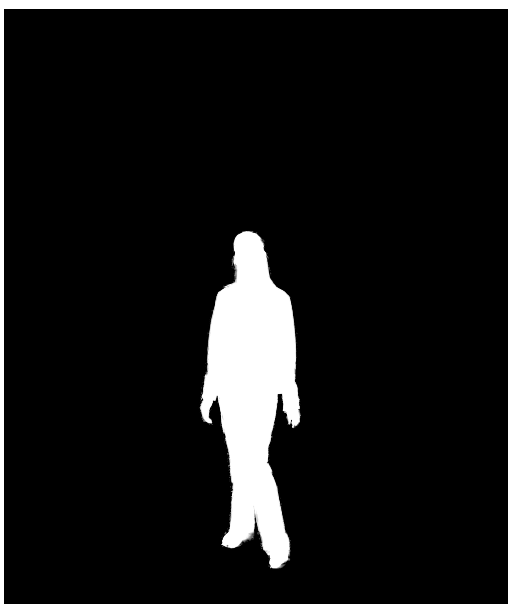


Figure 6.7: In this matting experiment, the proposed matting algorithm outperforms Poisson matting.

(a) given image

(b) given image



(c) extracted matte

(d) composite image

Figure 6.8: An application of the matting problem. The object of image (a) is extracted to combine with image (b). The composite image (d) looks realistic.

6.5 Conclusion

In this chapter, we propose to solve the matting problem by utilizing the inpainting algorithm presented in Chapter 5, which combines variational PDE-based inpainting and texture synthesis. We use the inpainting algorithm to extrapolate the foreground and background into the unknown region within the proposed iterative matting scheme. Our experiments show the proposed matting algorithm is effective for both geometric and texture images. In the future, we will improve the proposed matting algorithm so that it does not depend on the initial user-defined trimap. Several directions have been investigated as mentioned in introduction. One possibility is to add a region-growing step to generate a reasonable trimap from the user's rough guess. Another direction is to utilize the image data in the inpainting step that estimates the foreground and background in the unknown region, in addition to completing the geometry and texture.

REFERENCES

[1] Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. on Image Processing **10(2)** (2001) 266–277

[2] Kim, J., Fisher, J.W., Yezzi, A., Cetin, M., Willsky, A.S.: A nonparametric statistical method for image segmentation using information theory and curve evolution. IEEE Trans. on Image Processing **4** (2005) 1486–1502

[3] Herbulot, A., Jehan-Besson, S., Barlaud, M., Aubert, G.: Shape gradient for image segmentation using information theory. ICASSP **3** (2004) 21–24

[4] Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM Journal on Applied Mathematics **66(5)** (2006) 1632–1648

[5] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. Journal of Mathematical Imaging and Vision **28** (2007) 151–167

[6] Chan, T.F., Kang, S.H., Shen, J.: Euler's elastica and curvature based inpainting. SIAM journal on Applied Mathematics **63(2)** (2002) 564–592

[7] Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. ICCV **2** (1999) 1033–1038

[8] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int'l Journ. of Compt. Vision (1987) 321–331

[9] Cohen, L.: On active contour models and balloons. Compt. Vision Graphics Image Process. **53** (1991) 211–218

[10] Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int'l Journ. of Compt. Vision **22(1)** (1997) 61–79

[11] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Conformal curvature flows: from phase transitions to active vision. Archive for Rational Mechanics and Analysis **134** (1996) 275–301

[12] Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42** (1989) 577–685

[13] Osher, S., Sethian, J.A.: Front propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulation. J. Comput. Phys. **79** (1988) 12–49

[14] Zhu, S.C., Yuille, A.: Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **18** (1996) 884–900

[15] Yezzi, Jr., A.T.A., Willsky, A.: A statistical approach to snakes for bimodal and trimodal imagery. In Int'l Conf. on Computer Vision (1999) 898–903

[16] Paragios, N., Deriche, R.: Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision. Journal of Visual Communication and Image Prepresentation **13(1-2)** (2002) 249–268

[17] Jehan-Besson, S., Barlaud, M., Aubert, G., Faugeras, O.: Shape gradients for histogram segmentation using active contours. Proc. Int. Conf. Computer Vision (2003) 408–415

[18] Aubert, G., Barlaud, M.and Faugeras, O., Jehan-Besson, S.: Image segmentation using active contours: Calculus of variations or shape gradients? SIAM APPL. Math. **1(2)** (2005) 2128–2145

[19] Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33** (1962) 1065–1076

[20] Kantorovich, L.V.: On the translocation of masses. C.R.(Doklady) Acad. Sci. URSS(N.S.) **37** (1942) 199–201

[21] Villani, C.: Topics in optimal transportation. In: Graduate Studies in Mathematics. Volume 58. (2003)

[22] S.Rachev, L.Rüschendorf: Mass transportation problems. In: Vol.I: Theory, Vol.II: Applications.Probability and its applications. (1998)

[23] Chan, T., Esedoglu, S., Ni, K.: Histogram based segmentation using wasserstein distances. In: Proc. of Scale Space and Variational Methods in Computer Vision. Volume 4485/2007. (2007) 697–708

[24] Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces. Volume 153. (2002)

[25] Chambolle, A.: An algorithm for total variation minimization and applications. J. of Math. Imaging and Vision **20** (2004) 89–97

[26] Mory, B., Ardon, R.: Fuzzy region competition: A convex two-phase segmentation framework. in Proc. of SSVM (2007) 214–226

[27] Aujol, J.F., Gilboa, G., Chan, T.F., Osher, S.: Structure-texture image decomposition - modeling, algorithms, and parameter selection. Int'l Journ. of Computer Vision **67(1)** (2006)

[28] Rubner, Y., C.Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In IEEE Int'l Conf. on Computer Vision (1998) 59–66

[29] Chartrand, R., Vixie, K., Wohlber, B., Bollt, E.: A gradient descent solution to the monge-kantorovich problem. (submitted)

[30] Haker, S., Zhu, L., Tannnenbaum, A.: Optimal mass transport for registration and warping. Int'l Journ. of Computer Vision **60(3)** (2004) 225–240

[31] Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. Int'l Journ. of Compt. Vision **50(3)** (2002)

[32] Tu, Z., Zhu, S.C.: Image segmentation by data-driven markov chain monte carlo. IEEE Trans. on Pattern Analysis and Machine Intelligence **24(5)** (2002) 657–673

[33] Bresson, X., Chan, T.F.: Fast minimization of the vectorial total variation norm and applications to color image processing. UCLA CAM report 07-25 (2007)

[34] Julesz, B.: Visual pattern discrimination. IEEE Trans. on Information Theory **8** (1962) 84–92

[35] Julesz, B.: Textons, the elements of texture perception and their interatcions. Nature **290** (1981) 91–97

[36] Fogel, I., Sagi, D.: Gabor filters as texture discriminator. Journ. of Biological Cybernetics **61(2)** (1989) 103–113

[37] Chen, C.C., Dubes, R.C.: Discrete mrf model parameters as features for texture classification. IEEE Int'l Conf. on Systems, Man and Cybernetics **4(7)** (1990) 1–6

[38] Rubner, Y., Tomasi, C.: Texture-based image retrieval without segmentation. ICCV **2** (1999) 1018–1024

[39] Verma, M., Zisserman, A.: A statistical approach to texture classification from single images. IJCV **62(1-2)** (2005)

[40] Malik, J., Perona, P.: A computational model of texture segmentation. Proc. of CVPR (1989) 326–332

[41] Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. Int'l Journ. of Computer Vision **46(3)** (2002) 223–247

[42] Wolf, L., Huang, X., Martin, I., Dimitris, M.: Patch-based texture edges and segmentation. Proc. of ECCV 2006 **3952/2006** (2006) 481–493

[43] Zhu, S.C., Wu, Y.N., Mumford, D.: Filters, random fields and maximum entropy (frame). Int'l Journ. of Compt. Vision **27(2)** (1998) 1–20

[44] Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. Int'l Journ. of Computer Vision **40(1)** (2000) 1018–1024

[45] Zhu, S.C., Liu, X.W., Wu, Y.N.: Exploring julesz ensembles by efficient markov chain monte carlo–towards a trichromacy theory of texture. PAMI **22(6)** (2000)

[46] Wu, Y.N., Zhu, S.C., Liu, X.W.: Equivalence of julesz ensemble and frame models. IJCV **38(3)** (2000)

[47] Zhu, S.C., Wu, Y., Mumford, D.B.: Minimax etropy principles and its applications to texture modeling. Neural Computation **9** (1997) 1627–1660

[48] Zhu, S.C., Guo, C., Wang, Y., Xu, Z.: What are textons? IJCV **27(2)** (2005) 121–143

[49] Lindeberg, T.: Principles for automatic scale selection. In: Handbook on Computer Vision and Applications. Volume 2. (1999) 239–274

[50] Brox, T., Weickert, J.: A tv flow based local scale measure for texture discrimination. Proc. of ECCV (2004)

[51] Strong, D., Aujol, J., Chan, T.: Scale recognition, regularization parameter selection, and meyer's g norm in total variation regularization. UCLA CAM Report 05-02 (2005)

[52] Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica **D(60)** (1992) 259–268

[53] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int'l Journ. of Computer Vision **60(2)** (2004) 90–110

[54] Ni, K., Bresson, X., Chan, T.F., Esedoglu, S.: Local histogram based segmentation using the wasserstein distance. (in preparation)

[55] Huang, X., Qian, Z., Huang, R., Metaxas, D.: Deformable-model based textured object segmentation. EMMCVPR (2005)

[56] Bresson, X., Chan, T.F., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. J. of Math. Imaging and Vision **28(2)** (2007) 151–167

[57] Aujol, J.F., Gilboa, G., Chan, T.F., Osher, S.: Structure-texture image decomposition - modeling, algorithms, and parameter selection. IJCV **67(1)** (2006) 151–167

[58] Chung, J.T., Vese, L.A.: Image segmentation using a multilayer level-set approach. Technical Report CAM03-53 (2003)

[59] Lie, J., Lysaker, M., Tai, X.C.: A variant of the level set method and applications to image segmentation. AMS Mathematics of Computation **75** (2006) 1155–1174

[60] Brox, T., Weickert, J.: Level set segmentation with multiple regions. IEEE Trans. on Image Processing **15(10)** (2006) 3213–3218

[61] Sandberg, B., Kang, S.H., Chan, T.F.: Unsupervised feature balancing multiphase segmentation. Technical Report CAM08-02 (2008)

[62] Shashua, A., Ullman, S.: Structural saliency: the detection of globally salient structures using a locally connected network. International Conference on Computer Vision **5(8)** (1988) 321–327

[63] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. SIGGRAPH, Computer Graphics Proceedings (2000) 417–424

[64] Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. Proceedings of the International Conference on Computer Vision and Pattern Recognition (2001) 355–362

[65] Chan, T., Shen, J.: Mathematical models for local nontexture inpaintings. SIAM J. Visual Comm. Image Rep. **12** (2001) 436–449

[66] Bertozzi, A., Esedoglu, S., Gillette, A.: Inpainting of binary images using the cahn-hilliard equation. IEEE Trans. Image Proc. **60(2)** (2007) 285–291

[67] Bertozzi, A., Esedoglu, S., Gillette, A.: Analysis of a two-scale cahn-hilliard model for image inpainting. Multiscale Modeling and Simulation **6(3)** (2007) 913–936

[68] Esedoglu, S., Shen, J.: Digital inpainting based on the mumford-shah-euler image model. European J. Appl. Math. **13)** (2002) 353–370

[69] Dobrosotskaya, J.A., Bertozzi, A.: A wavelet-laplace variational technique for image deconvolution and inpainting. IEEE Trans. Imag. Proc. **17(5)** (2008) 657–663

[70] Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous texture and structure image inpainting. Proceedings of the International Conference on Computer Vision and Pattern Recognition (2003) 707–712

[71] Criminisi, A., Prez, P., Toyama, K.: Object removal by exemplar-based inpainting. Proceedings of the International Conference on Computer Vision and Pattern Recognition (2003) 721–728

[72] Grossauer, H.: Simultaneous texture and structure image inpainting. Proceedings of European Conference on Computer Vision (2004) 214–224

[73] Demanet, L., Song, B., Chan, T.: Image inpainting by correspondence maps: a deterministic approach. UCLA CAM Report 03-40 (2003)

[74] Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. SIGGRAPH, Computer Graphics Proceedings (2000) 479–488

[75] Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. Proc. of IEEE International Conference on Computer Vision **2** (1999) 1033–1038

[76] Ni, K., Roble, D., Chan, T.: A texture synthesis approach to elastica inpainting. ACM SIGGRAPH 2007 sketch program (2007)

[77] Smith, A.R., Blinn, J.F.: Blue screen matting. Proc. of SIGGRAPH (1996) 259–268

[78] Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. Proc. of CVPR **2** (2001) 264–271

[79] Ruzon, M.A., Tomasi, C.: Alpha estimation in natural images. Proc. of CVPR **1)** (2000) 18–25

[80] Sun, J., Jia, H., Tang, C., Shum, H.: Poisson matting. ACM Transactions on Graphics **23(3)** (2004) 315–321

[81] Berman, A., Vlahos, P., Dadourian, A.: Comprehensive method for removing from an image the background surrounding a selected object. U.S. Patent 6,124,345 (2000)

[82] Hillman, P., Hannah, J., Renshaw, D.: Alpha channel estimation in high resolution images and image seqences. Proc. of CVPR **1** (2001) 1063–1068

[83] Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. Proc. of IEEE CVPR (2007)

[84] Ni, K., Thiruvenkadam, S.R., Chan, T.F.: Matting through variational inpainting. Proc. of Signal and Image Processing (2007) 173–179

[85] Wang, J., Cohen, M.F.: An iterative optimization approach for unified image segmentaion and matting. Proc. of International Conference on Computer Vision (2005) 936–943

[86] Guan, Y., Chen, W., Liang, X., Ding, Z., Peng, Q.: Proc. of eurographics. IJCV **25(3)** (2006) 567–576

[87] Juan, O., Keriven, R.: Trimap segmentation for fast and user-friendly alpha matting. Proc. of Variational, Geometric, and Level Set Methods in Computer Vision (2005) 186–197

[88] Thiruvenkadam, S.R., Chan, T.F., Hong, B.W.: Variational image matting. (in preparation)