國立交通大學

多媒體工程研究所

碩士論文

基於結構差異性的影像分割

Image Segmentation Based on Structural Inconsistency

研 究 生：藍郁茜

指導教授：林奕成　教授

中華民國一百零二年八月

基於結構差異性的影像分割

Image Segmentation Based on Structural Inconsistency

研究生：藍郁茜　　Student：Yu-Chien Lan

指導教授：林奕成　　Advisor：I-Chen Lin

國立交通大學

多媒體工程研究所

碩士論文

A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年八月

摘要

我們提出一個新方法來處理影像分割問題,利用背景的結構一致性來區分前景。主要概念來自背景相減法。使用者只需要標示目標物體大略的邊框位置，我們的系統即可藉由最大化預測背景和真實背景的一致性，來找出物體的輪廓線。我們結合影像修補與前景分割的技術，使之融合為更有效的影像分割法。另外，我們也建立了一個新的影像資料庫，影像中的背景多為有結構性的物體，且前景物體顏色和背景顏色相似，難以判斷前景物體的輪廓線。利用我們的方法可有效切割出難以分辨的前景物體。

Abstract

We introduce a novel approach to deal with image segmentation which takes into account the consistent structure of the backgrounds. The concept is from background subtraction. Our method only needs users to specify their target objects by a bounding box. The system then finds the object contour by maximizing the consensus between the predicted background and the original image. We combine principles from image completion and foreground extraction approaches into a powerful unified engine. Besides, a new and harder dataset is introduced with images which have structural background objects.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

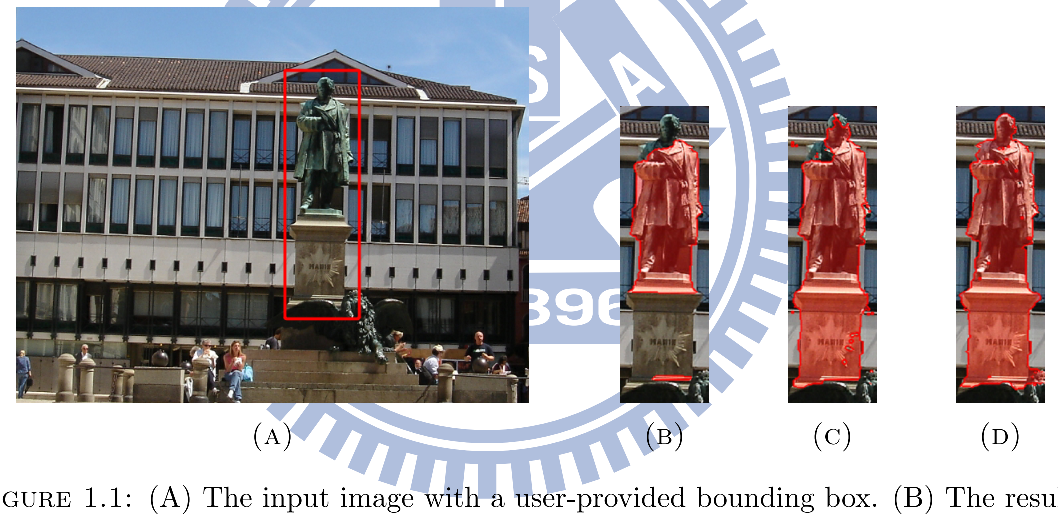


FIGURE 1.1: (A) The input image with a user-provided bounding box. (B) The result of GrabCut [Rother et al., 2004]. (C) The result of LVK [Lempitsky et al., 2009]. (D) Our result. Here is an example of the indistinguishable object and the results of three methods. By exploiting the structural background, our method obtains the best result (D).

Image segmentation techniques have been wildly developed nowadays. It is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. For instance, graph-based partitioning methods [Boykov and Jolly, 2001] tries to analyze the similarity between adjacent pixels. Image cosegmentation [Rother et al., 2006], exploiting the evidence from other images, segments objects in different images simultaneously by analyzing the coherence of the objects. Some other works extended the concept of image segmentation for video sequences [Friedman and Russell, 1997] or saliency map [Goferman et al., 2012].

Nevertheless, it is still challenging to automatically and precisely segment objects from images. In the domain of interactive image segmentation, they aim to use little user assistance for better segmentation results. Most of existing methods separate foregrounds and backgrounds by their color distribution and achieved impressive results in most cases. However, this strategy lacks ability to handle images whose color distribution of the desired object is quite similar to the background objects. We found that existing state-of-the-arts, such as GrabCut [Rother et al., 2004] or context-aware saliency map [Goferman et al., 2012] will fail to separate objects in such case. It is because these methods are driven by low-level stimulus such as intensity, color, orientation, and local texture. Without precise initial color of the foreground object, they will classify part or whole foreground region to the background. As a result, these methods have to ask for more user-intervention or require more obvious information about the objects. This situation, however, conflicts the demand of an automatic system.

Actually, we human beings distinguish an object not only the low-level but also higher level attributes. The structure of objects is a good feature for recognizing an object. However, we have not seen such an useful feature utilized in image segmentation researches. As a result, we develop a novel approach exploiting the consistent structure of backgrounds, and use the consensus to discriminate foregrounds. Besides, for automatic applications, it is easier for systems to indicate a bounding box than provide precise strokes within a target. Thus, we use merely one boundary box embraced the object as the user input and aim at accurately segmenting the foreground object.

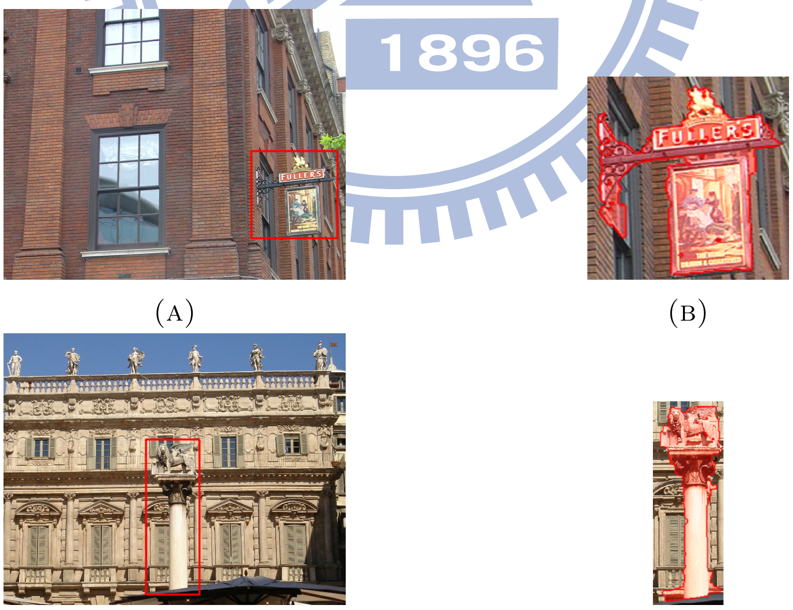
Chapter 2

Related Work

Our research is about image segmentation. We will briefly introduce recent state-of-the-arts in interactive image segmentation and image cosegmentation. Besides, our method can also be used in saliency detection. We will compare our research with saliency detection approaches. In addition, our background prediction method is based on image completion, we introduce several image completion articles as well.

Interactive image segmentation- Graph Cut, introduced by Boykov et al. [2001] partitions images into two parts based on the color distributions from user-indicated strokes. The concept is by treating each pixel in the image as a node in a graph and finding energy minimizing cuts in the graph. This method is efficient and effective and becomes a seminal core of many advanced methods. GrabCut [Rother et al., 2004], as an improved version of Graph Cut, allows a considerably reduced degree of user interaction. They employed Gaussian Mixture Model to approximate foreground and background probability distributions for digital matting. And instead of one-shot Graph cut, they proposed an iterative energy minimization. They used provisional labels on some pixels in the foreground which can subsequently be retracted in next iteration. This benefits to simple initial interactions. Users are allowed to use just a rectangle around the desired object instead of marking strokes on their targets. Lempitsky et al. [2009] further adds a bounding box prior into the GrabCut framework. This strategy prevents the solution from excessive shrinking and ensures that the user-provided box bounds the segmentation in a sufficiently tight way. Nieuwenhuis and Cremers [2013] considered not only the color distributions of user-provided strokes, but also the spatial distribution of the strokes. It can handle difficult images which exhibit strongly overlapping foreground and background color distributions due to large lighting variations.

Lazy Snapping [Li et al., 2004], an useful UI design for image cutout, proposes two steps: a quick object marking step and a simple boundary editing step. The first step specifies the object of interest by a few marking lines. The second step allows the user to edit the object boundary by simply clicking and dragging polygon vertices. This design is hard to handle thin and branch structures. Gulshan et al. [2010] demonstrated the power of shape constraints. This approach restricts the space of possible segmentations to a small subset and help to eliminate false segmentations. They manager to extend the notion of star-convexity from single to multiple centers in a traced way, and further generalized this notion from the Euclidean to geodesic.


(A)

(B)

(C)

(D)

Figure 2.1: Some examples of hard to segment objects and our segmented results. Many image segmentation methods will fail to detect regions like the stem of the billboard in (A), while our method obtains a good result.

Image cosegmentation techniques denote the task of simultaneously segmenting common objects of different images. Rother et al. [2006] presented a generative model with an MRF term encoding spatial coherency, and a global constraint which attempts to match the appearance histograms of the common parts. They demonstrated that supplying just one additional image can be sufficient to segment both together. Cosegmentation provides higher accuracy than that achieved with either one alone. Image cosegmentation techniques shows how useful it is if we give one more image to the segmentation system. From the other aspect, this also show the difficulty of precise single image segmentation.

Saliency detection techniques are used to detect the salient regions of an image. Goferman et al. [2012] proposes a novel algorithm for context-aware saliency detection. The idea is that salient regions are distinctive with respect to both their local and global surroundings. Although they try to use psychological evidence of human visual attention, their analyses are still regarding low-level considerations such as local texture and color. If the color of an object is similar to the surroundings (like the stem of the billboard in Figure 2.1), it will fail to detect this region out.

Image completion techniques, also referred to as inpainting or image filling, is an image editing tool for object removal and replacement or digital photograph restoration. It is used to fill holes after objects are removed. Criminisi et al. [2003] proposed an algorithm for removing large objects in images and replacing them with visually plausible backgrounds. They employ an exemplar-based texture synthesis technique modulated by a unified scheme for determining the fill order of the target region. It is capable of propagating both linear structure and two-dimensional texture into the target region. Shift-Map Image Editing [Pritch et al., 2009] described a new geometric rearrangement of images. They treated problems as an optimal graph labeling where the shift-map represents the selected label for each output pixel. Image Melding [Darabi et al., 2012] built upon a patch-based optimization foundation. They enriched the patch search space with additional geometric and photometric transformations, and integrated image gradients into the patch representation. With these improvements, it enables patch-based solutions to a broad class involving inconsistent sources. We find this technique is helpful for predicting background objects, and thus use the predicted background to distinguish foregrounds.

Chapter 3

Overview

We propose an iterative framework of image segmentation based on structural inconsistency. We first take image segmentation as an background subtraction problem. Unlike the traditional video-based background subtraction, we propose predicted the complete background by modified inpainting techniques. Given a user indicated bounding box, we assume the foreground object can be precisely extracted with perfect background filling. Therefore, we aim at reasonably filling the box region by known background regions. With this assumption, we can get a predicted background and can further be used to distinguish possible foreground locations. We call this step as ”Image segmentation using structural inconsistency”. At this step, we will get a rough object contour.

The result obtained in the structural-inconsistency-based subtraction is a indicating mask with the approximate foreground object position and silhouette. To further refine the contour, we manager to use the approximate contour to generate strokes like user interaction of the foreground extraction method. By means of these strokes, we use a graph-cut-based foreground extraction tool for a more precise segment. We use *Geodesic Star Convexity* [Gulshan et al., 2010] at this step because of its content of shape constraint. We will introduce each step in detail in the following chapters. Figure 3.1 is the flow chart of the proposed framework.

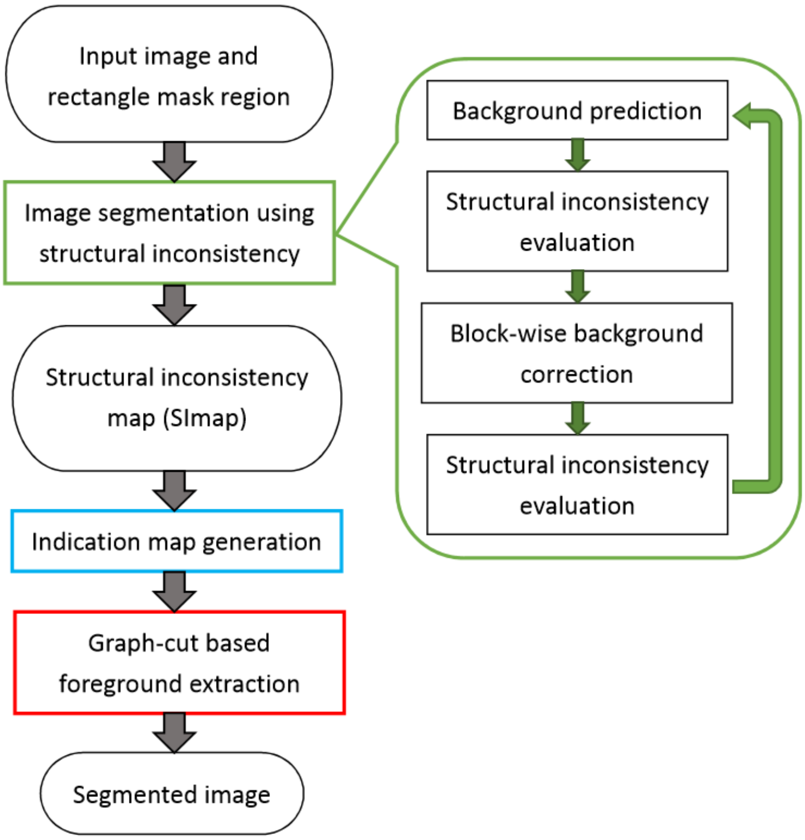

Input image and rectangle mask region
Image segmentation using structural inconsistency
Structural inconsistency map (SImap)
Indication map generation
Graph-cut based foreground extraction
Segmented image
Background prediction
Structural inconsistency evaluation
Block-wise background correction
Structural inconsistency evaluation

FIGURE 3.1: The flow chart of the proposed system.

Our system has three main steps:

1. **Image segmentation using structural inconsistency-** we first apply image completion method to predict the background in the user-indicated rectangle region. We use *Image Melding* [Darabi et al., 2012] as the essential inpainting method. While we have predicted backgrounds, we can examine the similarity of the predicted backgrounds with the original image. By this evaluation, we understand which region may be the background regions and thus can obtain possible foreground regions. This method is denoted as ”Structural inconsistency evaluation”. However, the inpainted regions may have distortion compared with the ”true” background, so we further apply ”Block-wise background correction” to the inpainted regions. We divide the inpainted regions into blocks and transform each block geometrically for correctness. And again, we apply ”Structural inconsistency evaluation” to the

corrected background. Finally, we can get a ”Structure inconsistency map (SImap)” indicating possible foreground regions.

2. **Indication map generation-** we generate an indication map by using the SImap. Since the contour of SImap is only approximated through structural inconsistency, we would like to further refine the border by graph-cut-based methods. Therefore, we construct a more conservative map indicating where are the true foreground locations and where are the true background locations. The constructed map can be regarded automatic-predicted strokes (labels) for Graph-cut optimization in the next step.

3. **Graph-cut based foreground extraction-** by the indication at the previous step, we can further take it on a foreground extraction tool for more precise segmentation. With the helpful labels in the indication map, we can get final precise segmented results.

Chapter 4

Methods

Existing interactive image segmentation methods may fail in images whose color of foreground is quite similar to the color of backgrounds. For these images, they will ask users provide more cues about the object locations, such as marking more precise boundary positions of the desired objects. This situation, however, conflicts the objective of an intelligent segmentation system. Therefore, we intend to provide a system that only need the initial bounding box embraced the object, and with no more interactions, our system can then automatically segment the object more precisely than existing methods. It can greatly decreases user interactions and benefits to automatic computing systems.

We now introduce our algorithm of image segmentation by structural inconsistency. There are three main steps: Image Segmentation Using Structural Inconsistency (Sec 4.1), Indication Map Generation (Sec 4.2), Graph-Cut-Based Foreground Extraction (Sec 4.3).

4.1 Image Segmentation Using Structural Inconsistency

Image inpainting or image completion is pervasive used as a restoration tool of images nowadays. We found the ability of inpainted image can be used for extracting foreground objects by treating the inpainted region as predicted backgrounds. Nevertheless, straightforwardly using background subtraction with inpainted images does not have satisfactory results. Therefore, we proposed a consistency matching framework. Our framework in this step goes through three parts: background prediction, structural inconsistency evaluation and block-wise background correction. We conduct an iterative process to find a coarse boundary contour of the foreground object by using the characteristics of structural consistency. For each iteration, we first inpaint the masked region specified by users, and then subtract the inpainted image from the original source image to obtain a structural inconsistency map (SImap). By examining each pixel value in the SImap from the boundary of the mask region to the center and discarding pixels under a particular threshold, we can get a binary mask indicating possible foreground region. Finally, we take this binary mask as the input of the first step. Then our system iteratively removes background regions until the binary mask is no longer changed or reaches a particular number of iterations. The inpainting regions will gradually become smaller in each iteration, which means, the source region for inpainting will become bigger offering more clues of background information and resulting in a more accurate inpainting image (Figure 4.1). We include pseudo-code of our algorithm in Algorithm 1, and the details are described in the following subsections.

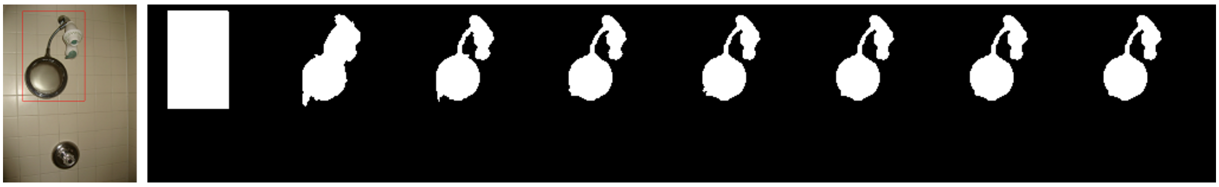


FIGURE 4.1: An example of the changing of foreground regions in each iteration with the first image as the input.

Algorithm 1 Framework of Image Segmentation Using Structural Inconsistency

```
Input: Input image $S$ and bounding box mask $T$
Output: Binary mask $B$ indicating foreground region
Downscale $S$ and mask $T$
$m = T$
for iteration $i = 1 \to n$ do
    $\Omega \leftarrow \text{BackgroundPrediction}\ (S, m)$
    $D = \|\Omega - S\|$
    $t \leftarrow \text{PredictDiffThreshold}(D, m)$
    $m \leftarrow \text{StructuralInconsistencyEvaluation}(D, t, m)$
    $\Omega' \leftarrow \text{BackgroundCorrection}(\Omega, S)$
    $D' = \|\Omega' - S\|$
    $m \leftarrow \text{StructuralInconsistencyEvaluation}(D', t, m)$
    if IsConverged($m$) then
        break
    end if
end for

Upscale $m$
$B = m$
```

4.1.1 Background Prediction

We choose *Image Melding* [Darabi et al., 2012] as our inpainting tool since it outperforms previous state-of-the-art methods in the field of image inpainting. *Image Melding* builds upon a patch-based optimization foundation.

Given a user-defined rectangle mask, the input image is then divided into the source region S and the target region T (Figure 4.2). The objective is to replace content of T with content from region S. They suggest a patch-based optimization problem by minimize the following function:

$$E(T, S) = \sum_{q \subset T} \min_{p \subset S} \left(D(Q, P) + \lambda D(\nabla Q, \nabla P)\right), \tag{4.1}$$

where $Q = N(q)$ is a $\omega \times \omega$ patch with target pixel q at its upper left corner, and $P = f(N(p))$ is a $\omega \times \omega$ patch that is a result of a geometric and photometric transformation

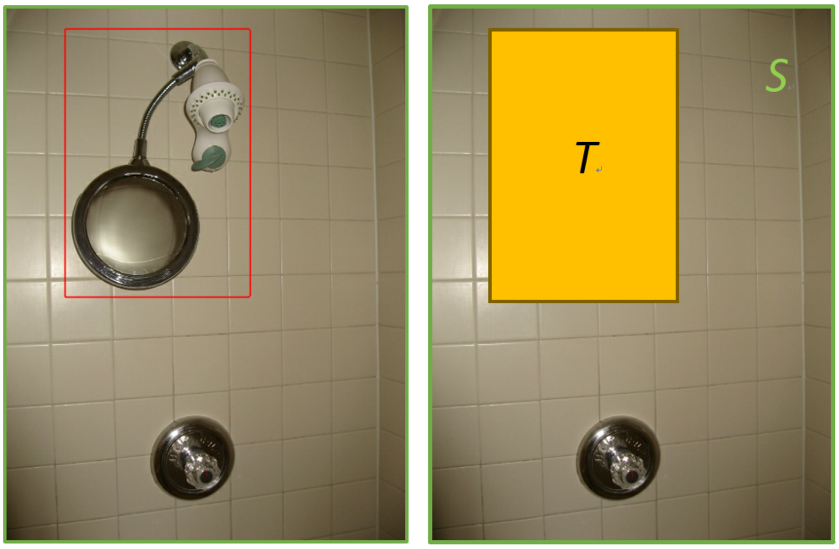


FIGURE 4.2: Left: the input image image with user provided bounding box. Right: the target region T and the source region S for image inpainting.

f applied on a small neighborhood N around source pixel p. P and Q has three color channels. ∇P and ∇Q are the two luminance gradient channels on source and target region respectively. The above energy function finds a cover of the missing data using the available one. That is, to find the optimal patches which are the most similar to the local neighborhood in the source region. With this method, we can get a background prediction for separating the different structure in the foreground (Figure 4.3).

4.1.2 Block-wise Background Correction and Structural Inconsistency Evaluation

Even though the predicted background is highly similar to the original region, there may be still slight unfitness between the predicted background and the original one. Thus, we further apply local image warping to the inpainted region. We divide the inpainted region into 15×15 blocks. For each block, our system finds the best affine transformation to the block in the original image. It is denoted as ”Block-wise Background Correction”. The minimization function aim to find the best affine matrix of predicted background and original image:

$$\arg\min_{\mathbf{A}} \sum_{i} |I(\mathbf{A}\mathbf{q_i}) - I(\mathbf{p_i})|^2, i \in \text{block region} \tag{4.2}$$

$$\text{subject to } |\mathbf{A} - \mathbf{I}|^2 < \text{threshold } k \tag{4.3}$$

where A is the affine transformation matrix, $I(\mathbf{p})$ gets the color value in position $\mathbf{p}$, and q is in predicted background image, p is in original source image. I is the identity matrix. With a constraint of matrix $\mathbf{A}$, it bounds the transformation in a small set preventing big deformation. We use *Levenberg-Marquardt* algorithm to find the best affine transformation. In this way, each inpainted block will deform to a more precise fitting to the ground truth input image.

As long as we have the predicted background image, we can evaluate the inconsistency between source image and predicted backgrounds. We call this step as ”Structural Inconsistency Evaluation”. We can obtain a Structural Inconsistency map (SImap) by subtracting predicted background from source image (Figure 4.5).

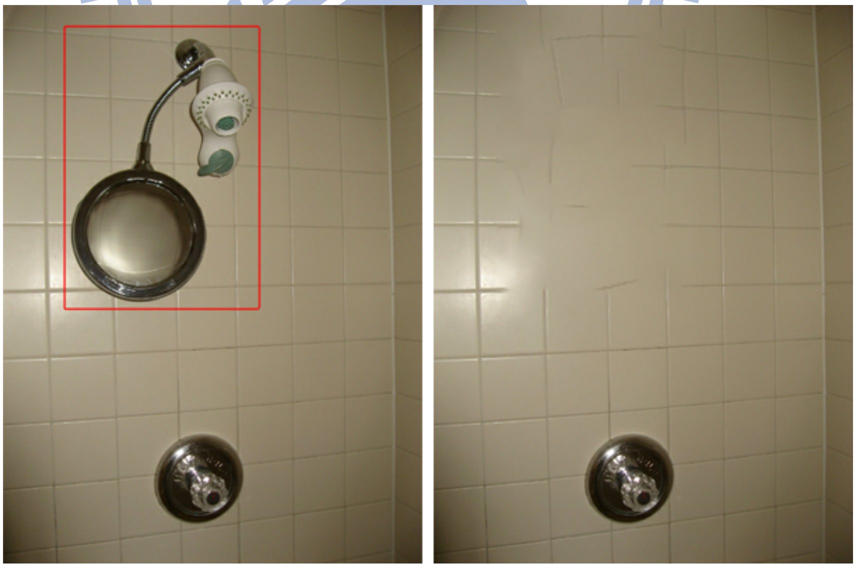


Figure 4.3: An example of background prediction result. Left: input image with a bounding box specified by users. Right: the inpainted result.

$$D(i) = \|\Omega(p_i) - S(p_i)\|, \tag{4.4}$$

where i is pixels belonged to source image, Ω is the inpainted image with best warping, and S is the source image. D collects the L1 norm distance of each pixel. Then, we check each pixel in D inwardly from the border of the rectangle mask.

$$Seg(i) = \begin{cases} 1 & \text{, if } D(i) > t; \\ 0 & \text{, otherwise.} \end{cases} \tag{4.5}$$

$Seg(i)$ is a binary mask gathering the pixels whose value is above a threshold t. The notable strategy is that we check each pixel inwardly from the mask border, until there is a boundary contour whose values of pixels belonged to the contour are above the threshold t. A concept demonstration is showed in Figure 4.4. The threshold t is predicted as the value at the last 10% of difference distribution in mask region.

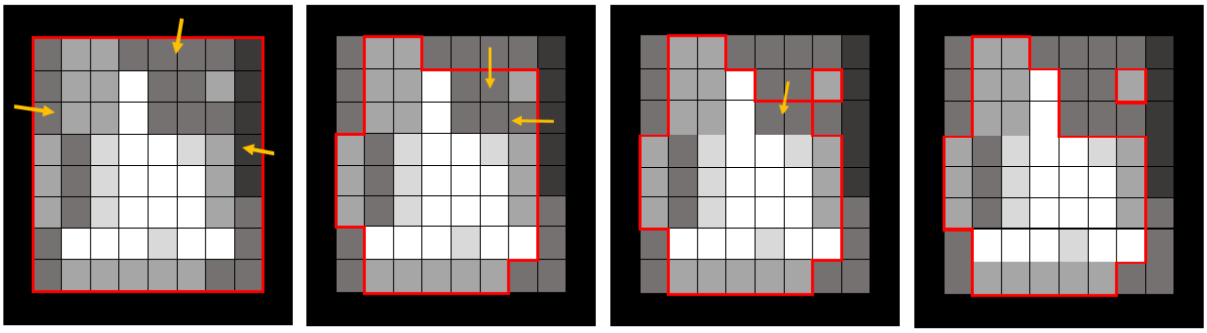


FIGURE 4.4: Inward structural inconsistency checking. The process starts from the mask boundary to the center in sequence and converges until there is a boundary blocked out checking process. The pixel values outside the boundary region are all under the threshold t.

4.1.3 Refinement

In this step, we further manipulate the $Seg(i)$ image for a better result. First, we use majority to the binary mask, a morphological operation which sets a pixel to 1 if five or more

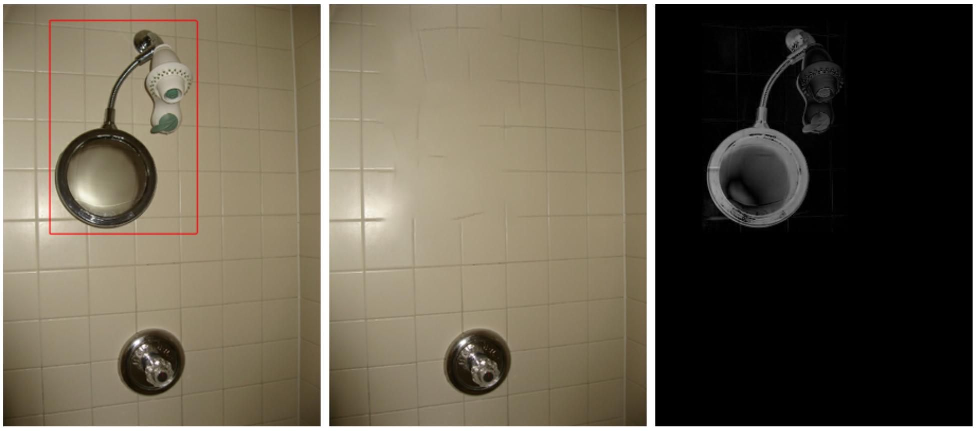


FIGURE 4.5: An example of structural inconsistency map (right) produced by subtracting inpainted image (middle) from source image (left).

pixels in its 3-by-3 neighborhood are 1s; otherwise, it sets the pixel to 0. This operation attempts to centralize the mask helping to get a compact boundary contour. Second, We find the biggest connected component of the binary image eliminating unconnected parts.

Our system iteratively executes three steps introduced in section 4.1.1 to 4.1.3. Our objective is to gradually shrink the possible foreground regions, in other words, to reduce the essential background prediction regions. This strategy takes the advantages of image completion results since the image completion is good at borders within the inpainting regions. And therefore, we use the inwardly structural inconsistency evaluation as described in section 4.1.2. For each pixel, the larger its distance away from inpainting boundary, the less information it can receive about the surrounding objects. Compared with the one-shot procedure by executing background prediction one time and then tried to analyze the only SImap, we found that our iterative procedure can gradually approach the object silhouette and lessen the effects of imperfect inpainting results.

4.2 Indication Map Generation

This step intend to generate an indication map denoting foreground strokes and background strokes for graph-cut optimization. As we have obtained a coarse binary mask of the foreground object, it can be used to create strokes for a more precise segmentation. At first, we take the boundary contour C_s of the binary mask Seg. Then, by applying dilation operation on C_s, we can get a band shape of thicker contour. This band has two boundaries: one inside the mask region of Seg, and the other one outside. We then treat the outside boundary as the background strokes, and inside boundary as the foreground strokes (Figure 4.6).

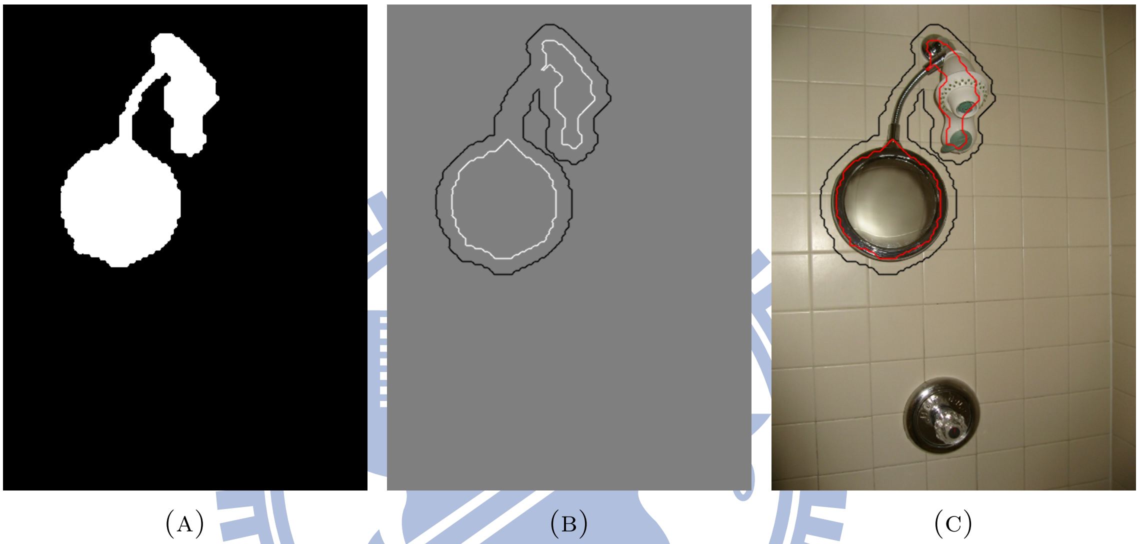


FIGURE 4.6: (A) The final mask obtained from Sec 4.1. (B) The Indication Map. (C) A demonstration of locations of foreground strokes (Red) and background strokes (Black).

4.3 Graph-cut-based Foreground Extraction

We have tried various foreground extraction tools [Boykov and Jolly, 2001, Grady, 2006, Liu et al., 2009] and find the method of *Geodesic Star Convexity* [Gulshan et al., 2010] gets the best result with our indication map. We take the indication map as the input of *GSC*. With the shape constraint introduced by *GSC*, we obtain a precise segmentation at this final step. Insufficient segmented boundary can be improved at this step (Figure 4.7).

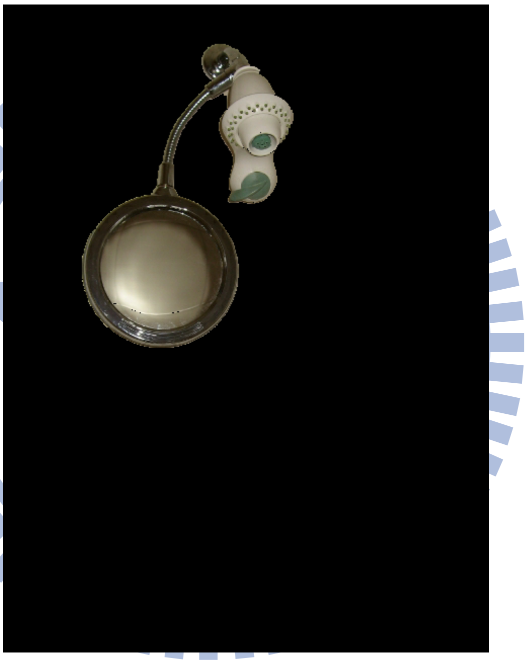


FIGURE 4.7: The result of *GSC* by our indication map (Figure 4.6).

Chapter 5

Experiments

5.1 Quantitative Evaluation

We compare the average quality of our method to the previous methods, Grabcut [Rother et al., 2004] and LVK [Lempitsky et al., 2009]. Both of these two methods can be initialized by one user-provided bounding box. Without further user interaction, our method outperforms these methods and provide more precise segmentation results.

We construct a database with 80 images which have obvious structural backgrounds. Some images are difficult to be segmented because the color of foreground objects is quite similar to the background objects. We took 72 images from LabelMe dataset [Russell et al., 2008] with ground truth marked by users. And 8 images are from Grabcut database with ground truth provided by the authors.

5.1.1 Evaluation

We utilize the two well-known measures precision and recall and F_2-measure to evaluate the accuracy of comparative methods. The precision and recall is defined by

$$Precision = \frac{tp}{tp + fp} \tag{5.1}$$

$$Recall = \frac{tp}{tp + fn} \tag{5.2}$$

where tp is the set of correct results, fp is the set of unexpected results, and fn is the set of missing results. The score is measure by F_2-measure.

$$F_2 = \frac{5 \cdot Precision \cdot Recall}{4 \cdot Precision + Recall}. \tag{5.3}$$

Table 5.1 shows the F_2-measure of our method together with Grabcut [Rother et al., 2004] and LVK [Lempitsky et al., 2009]. Our method achieve the highest score in this dataset. Figure 5.1, 5.2 and 5.3 show some examples of the segmentation results. We can see many objects are difficult to distinguish out. But our method performs the highest quality segmentation results.

Method	F_2-measure
Our method	0.9594
LVK	0.9207
Grabcut	0.8713

TABLE 5.1: F_2-measure scores in our structural dataset.

5.1.2 Comparison on Other Dataset

In order to prove the generality of our approach, we further test our method on *Complex Scene Saliency Dataset* [Yan et al.]. Because our algorithm depends on background prediction by image completion, we can not handle objects without sufficient background information. Therefore, we choose 50 out of 200 images as the testing data. We demonstrate the F_2-measure scores and compare with Grabcut [Rother et al., 2004] and LVK

[Lempitsky et al., 2009] in Table 5.2. It shows that our performance is as good as LVK method. And it confirms the generality of our approach even though the backgrounds are not structural objects. Results are showed in Figure 5.4.

Method	F_2-measure
Our method	0.9267
LVK	0.9248
Grabcut	0.8928

Table 5.2: F_2-measure scores in Complex Scene Saliency Dataset.

5.2 Discussions

We mainly compare our method to Grabcut and LVK. Indeed, Grabcut provides an efficient, iterative version of graph-cut-based optimization. They use Gaussian Mixture Models for color data modelling and estimating the probability of each pixel. The iterative energy minimization benefits to improve the parameters of GMMs. This method is good at images with simple backgrounds and objects of high contrast color distributions. Following from the concept of Grabcut, LVK further adds a bounding box prior to the Grabcut framework. They propose an algorithm helping for finding a tight shape. The shape should satisfy the tightness of user-provided bounding box. Nevertheless, LVK can not understand the real shape of an object. They can only find a conservative shape which can satisfy the tightness constraint. As a result, if the target object has a similar color distribution to the background, the approach of LVK will fail to segment in this situation. However, our method takes the advantage of consistent background structure, and recognizes the shape of an object by the distinct structure from backgrounds. Our method can segment the desired object with high accuracy.

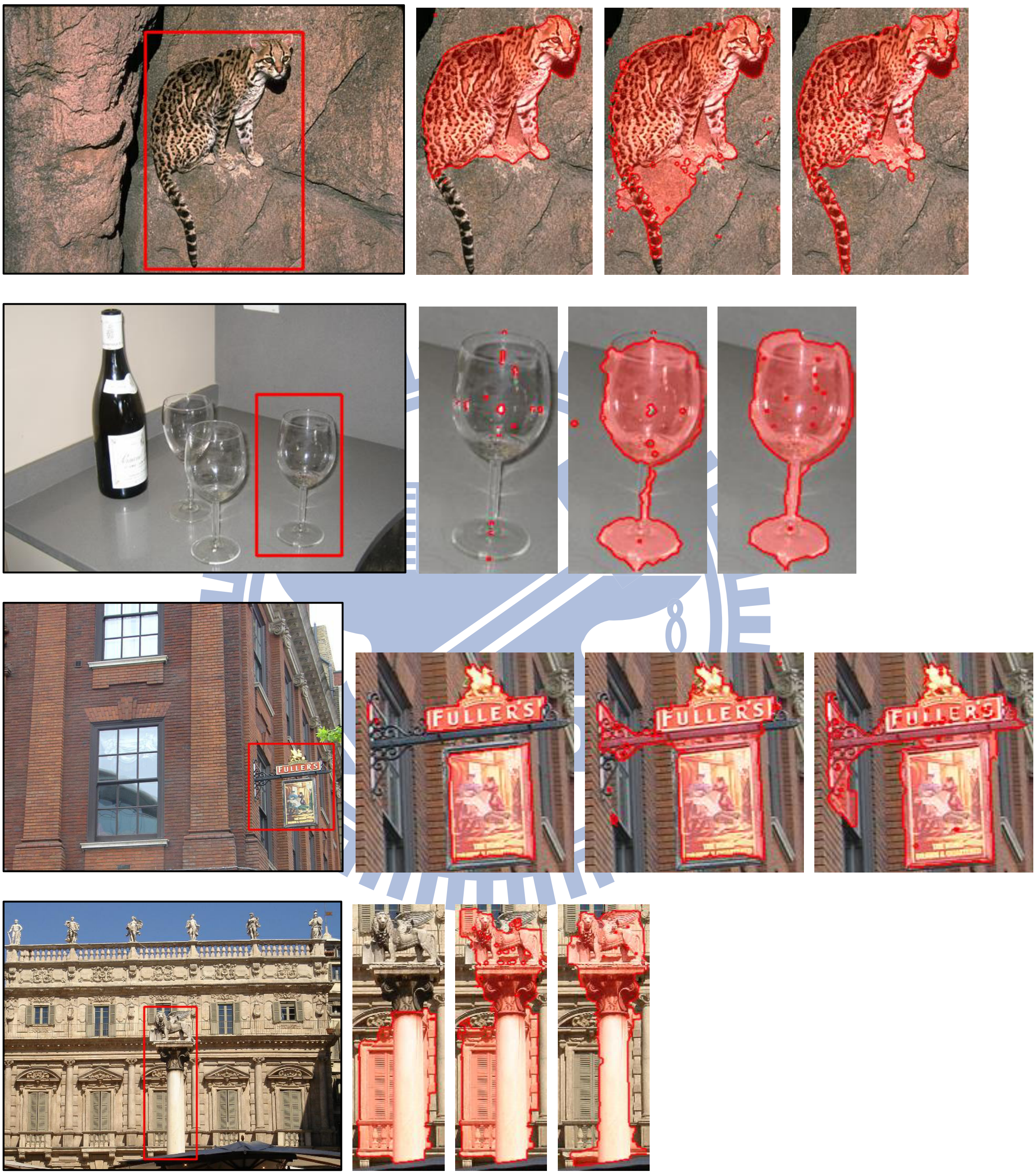


Figure 5.1: Results from our structural dataset. From left to right: (1) The input image. (2) The results of Grabcut. (3) The results of LVK. (4) Our results.

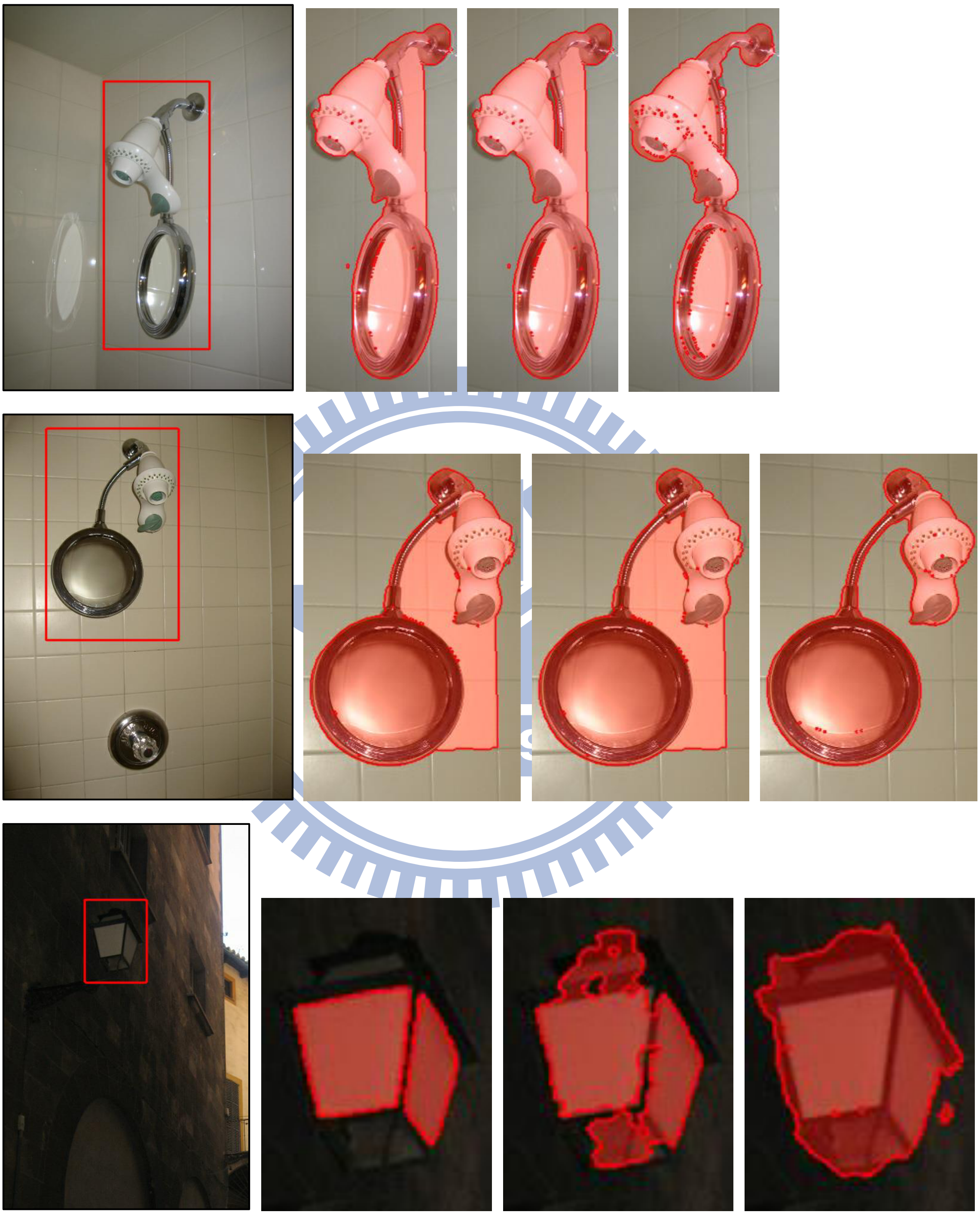


FIGURE 5.2: Results from our structural dataset. From left to right: (1) The input image. (2) The results of Grabcut. (3) The results of LVK. (4) Our results.

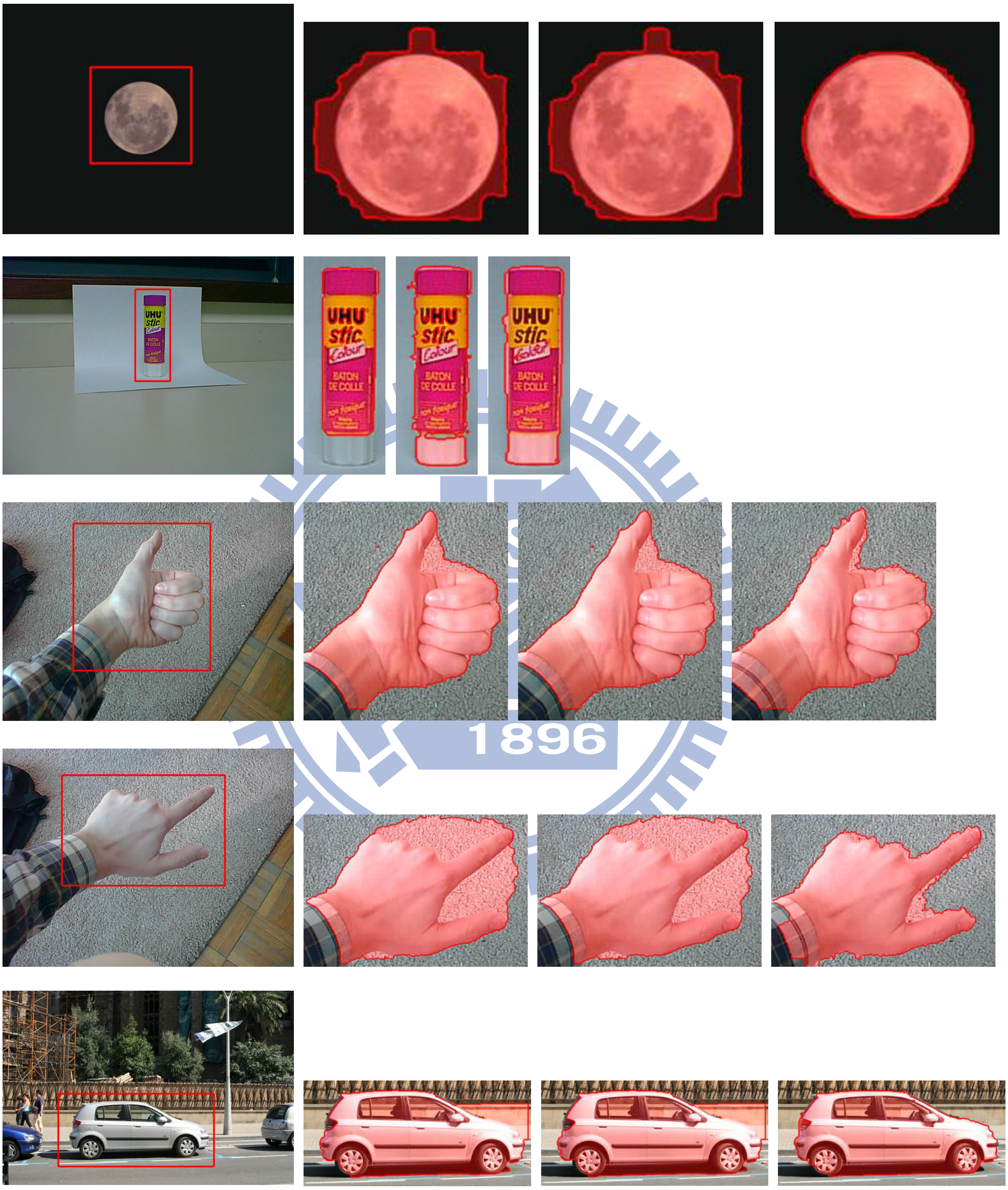


Figure 5.3: Results from our structural dataset. From left to right: (1) The input image. (2) The results of Grabcut. (3) The results of LVK. (4) Our results.

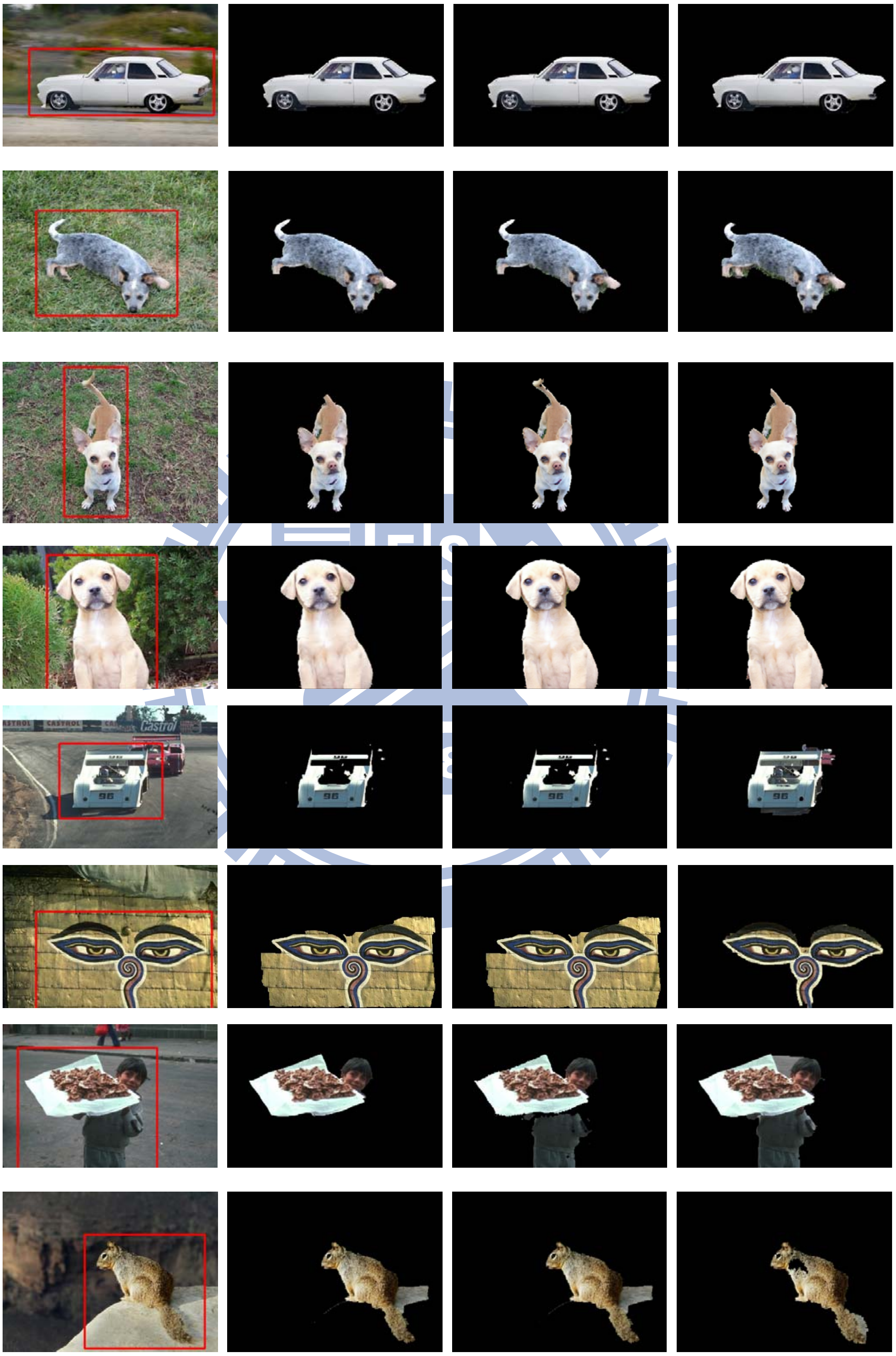


Figure 5.4: Results from the CSSD dataset. From left to right: (1) The input image. (2) The results of Grabcut. (3) The results of LVK. (4) Our results.

Chapter 6

Conclusions and Limitations

We have shown a structure-based image segmentation framework that use structure consensus for foregrounded and background separation. Our method combines principles from image completion and foreground extraction approaches into a powerful unified engine. And we demonstrate the high precision of our method in indistinguishable images by only one box as input. Our method can greatly decrease user interaction and benefits to further automatic segmentation systems.

Our method still has a few limitations: the input image should content sufficient background information. And if there is another object identical to the target object in the image, our system may regard the object as parts of background and will fail in this situation.

Bibliography

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015720. URL http://doi.acm.org/10.1145/1015706.1015720.

Victor S. Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, pages 277–284. IEEE, 2009. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2009.html#LempitskyKRS09.

Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1, 2001. doi: 10.1109/ICCV.2001.937505.

Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 993–1000, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.91. URL http://dx.doi.org/10.1109/CVPR.2006.91.

Nir Friedman and Stuart Russell. Image segmentation in video sequences: a probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, UAI'97, pages 175–181, San Francisco, CA, USA, 1997. Morgan Kaufmann

Publishers Inc. ISBN 1-55860-485-5. URL http://dl.acm.org/citation.cfm?id=2074226.2074247.

Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1915–1926, 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.272.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. URL http://dblp.uni-trier.de/db/journals/pami/pami23.html#BoykovVZ01.

C. Nieuwenhuis and D. Cremers. Spatially varying color distributions for interactive multi-label segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1234–1247, 2013. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.183.

Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015719. URL http://doi.acm.org/10.1145/1015706.1015719.

V. Gulshan, C. Rother, Antonio Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3129–3136, 2010. doi: 10.1109/CVPR.2010.5540073.

A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–721–II–728 vol.2, 2003. doi: 10.1109/CVPR.2003.1211538.

Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg. Shift-map image editing. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 151–158. IEEE, 2009. doi: http://dx.doi.org/10.1109/ICCV.2009.5459159.

Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. Image melding: combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.*, 31(4):82, 2012. URL http://dblp.uni-trier.de/db/journals/tog/tog31.html#DarabiSBGS12.

L. Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.233.

Jiangyu Liu, Jian Sun, and Heung-Yeung Shum. Paint selection. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 69:1–69:7, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-726-4. doi: 10.1145/1576246.1531375. URL http://doi.acm.org/10.1145/1576246.1531375.

Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77 (1-3):157–173, May 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0090-8. URL http://dx.doi.org/10.1007/s11263-007-0090-8.

Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Complex scene saliency dataset. http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

Connelly Barnes, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV (3)*, volume 6313 of *Lecture Notes in Computer Science*, pages 29–43. Springer, 2010. ISBN 978-3-642-15557-4. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2010-3.html#BarnesSGF10.

Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):463–476, 2007. URL http://dblp.uni-trier.de/db/journals/pami/pami29.html#WexlerSI07.

Johannes Kopf, Wolf Kienzle, Steven M. Drucker, and Sing Bing Kang. Quality prediction for image completion. *ACM Trans. Graph.*, 31(6):131, 2012. URL http://dblp.uni-trier.de/db/journals/tog/tog31.html#KopfKDK12.

Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 313–318, New York, NY, USA, 2003. ACM. ISBN 1-58113-709-5. doi: 10.1145/1201775.882269. URL http://doi.acm.org/10.1145/1201775.882269.

Hui Fang and John C. Hart. Detail preserving shape deformation in image editing. *ACM Trans. Graph.*, 26(3):12, 2007. URL http://dblp.uni-trier.de/db/journals/tog/tog26.html#FangH07.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3), 2009. URL http://dblp.uni-trier.de/db/journals/tog/tog28.html#BarnesSFG09.

Yu-Shuen Wang, Chiew-Lan Tai, Olga Sorkine, and Tong-Yee Lee. Optimized scale-and-stretch for image resizing. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, pages 118:1–118:8, New York, NY, USA, 2008. ACM. ISBN 978-1-4503-1831-0. doi: 10.1145/1457515.1409071. URL http://doi.acm.org/10.1145/1457515.1409071.

Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882264. URL http://doi.acm.org/10.1145/882262.882264.

Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, November 1998. ISSN 0162-8828. doi: 10.1109/34.730558. URL http://dx.doi.org/10.1109/34.730558.

Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24(3):795–802, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073263. URL http://doi.acm.org/10.1145/1073204.1073263.