國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

以非剛性對齊演算法與使用者標記之明顯特徵對應關係建立影片資料之稠密對應

Dense Correspondence Annotation of Video Data Using Non-Rigid Registration with Salient Feature Correspondence Constraints

陳彥廷

Yen-Ting Chen

指導教授：王傑智 博士

Advisor: Chieh-Chih Wang, Ph.D.

中華民國 一百零三 年 十一 月

November, 2014

國立臺灣大學碩士學位論文
口試委員會審定書

以非剛性對齊演算法與使用者標記之明顯特徵對應關係建立影片資料之稠密對應

Dense Correspondence Annotation of Video Data Using Non-Rigid Registration with Salient Feature Correspondence Constraints

本論文係陳彥廷君（學號R01922116）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 103 年 10 月 8 日承下列考試委員審查通過及口試及格，特此證明

口試委員：王傑智

（指導教授）

康里力　簡忠漢

系主任　趙坤茂

誌謝

轉眼間在實驗室已經待了兩年多，感謝王傑智老師的循循善誘，引領我進入學術研究的殿堂，讓我了解什麼是做研究的態度及做研究的方法。最重要的是我學習到如何將一件事情確實的表達給別人，不論是寫文章或是實際與人面對面的互動。我很欣賞王老師對研究的熱情及執著，他總是充滿自信，以非常快的反應速度講出切中要點的話。不論是做研究或是做事的方法及態度，我從他身上學習到很多我值得去努力的事情。

感謝實驗室的學長、同學及學弟們花費時間及精神幫我測試系統，收集資料及給我建議。你們的幫助讓我的實驗結果更加豐富，也讓我了解系統架構有哪些需要改進的地方。平時大家在實驗室總是打成一片，遇到困難的時候總是會互相幫忙。這深深地讓我覺得我們就是一個團隊，可以一同歡笑，可以一起完成很多困難的任務。

感謝我的家人和女朋友總是支持及鼓勵我，有時縱使我表現不好，他們也從不責怪我，只希望我能從失敗中汲取教訓，繼續努力。在他們的支持及鼓勵下我更能無後顧之憂地去完成我的學業，我真的很感謝他們。

感謝我有機會在台大求學，這裡的環境讓我感到非常舒適。學校的系所非常的多，有來自世界各地的老師及學生，接觸各式各樣的人讓我覺得自己的視野又開闊了一點，心胸也變得更寬大，能夠接受更多不一樣的思考方式。此外，對一個住宿生而言，學校附近有很多店家可以滿足各種生活需求且又緊鄰商圈及捷運站，生活十分地便利。

最後，期許自己能夠繼續努力，將自己這兩年多來所學加以精進，更加充實自己，希望有朝一日可以為這個社會盡一點心力。

摘要

大部分現有的影片標記系統(video annotation system)專注於標記影片中物體的行為(activity)，其他的系統則是致力於標記出影片中每個物體的位置甚至是物體的輪廓(object contour)。我們發現後者只利用定界框(bounding box)或是利用內插法幫助使用者標記一個物體在每個幀(frame)中對應的位置或輪廓，而其中只有一篇著作提及如何去找出被標記的物體之中的稠密對應關係(dense correspondence)。經過分析之後我們發現影片資料之稠密對應關係標記還有許多議題需要釐清。因此，我們發展了一個標記對應物體輪廓之中每個像素的對應關係的影片標記系統。

此外，由於標記整個影片中物體的細部輪廓以及稠密對應關係必須花費許多精神和時間，我們利用互動式分割(interactive segmentation)、光流法(optical flow)及邊緣檢測(edge detection)的結果讓使用者可以更容易觀察出兩個幀之間的明顯特徵對應關係(salient feature correspondence)。邊緣檢測的結果可以幫助使用者找出物體的細部輪廓或是物體局部的圖樣(local pattern)。我們要求使用者確認及修改演算法找出來的明顯特徵對應關係。而對於物體中沒有特徵的區域(textureless region)，我們將使用者標記在兩個相鄰幀的明顯特徵對應關係做非剛性對齊(non-rigid registration)來得到此區域的稠密對應關係。使用者只需要仔細的標記第一個幀的物體輪廓及明顯特徵然後再修正演算法錯誤的部分就可以將整個影片標記完成。實驗結果顯示我們的系統較適合用來標記非剛性的物體。

DENSE CORRESPONDENCE ANNOTATION OF VIDEO DATA USING NON-RIGID REGISTRATION WITH SALIENT FEATURE CORRESPONDENCE CONSTRAINTS

Yen-Ting Chen

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

November 2014

Submitted in partial fulfilment of
the requirements for the degree of
Master of Science

Advisor: Chieh-Chih Wang

Thesis Committee:
Feng-Li Lian
Jong-Hann Jean (St. John's University)

ABSTRACT

THERE are a few existing annotation systems that aim to provide a platform for video annotation. Most of them focus on activity annotation while others concentrate on labeling individual objects. However, the latters focus on only labeling objects with bounding boxes or only using interpolation techniques to help user labeling. Moreover, only one of them try to find the dense correspondence inside the object contour. Issues of dense correspondences annotation across video frames are not well addressed yet. Inspired by this, a video annotation system that focuses on dense correspondences annotation inside the object contour is proposed in this work. In addition, since labeling detail object contour and dense correspondences across a whole video is a daunting task, we also minimize user's effort by applying an interactive segmentation and tracking algorithm that utilizes information from optical flow and edges that helps the user easier to observe the salient feature correspondences between two video frames. Edges could help the user to find out the detail contour or local patterns of the object. The user is required to check and modify the salient feature correspondences obtained by the algorithm. Dense correspondences in the textureless region are extracted by a non-rigid registration algorithm from the salient feature correspondences verified by the user. The user only needs to label the first frame of the video and correct some minor errors in the subsequent frames for the whole video annotation. The result shows that the proposed framework is more suitable to label non-rigid objects.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

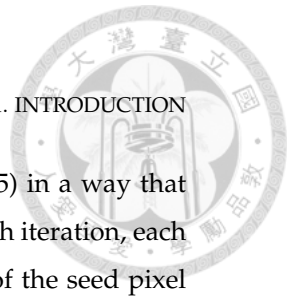
CHAPTER 1

Introduction

SEMANTIC segmentation, object recognition, and moving object detection are critical in the robotics and computer vision literature. In order to evaluate the performance, an efficient ground truth annotation system is essential. Based on different sensor types, the user is requested to label laser range points data, RGB data, or RGBD data which an annotation system needs to deal with. To ease the effort of labeling, different annotation systems and algorithms are developed. Laser range points annotation system is proposed in (Weng et al., 2012). RGB image annotation systems are proposed in (Russell et al., 2008) (Giro-i Nieto et al., 2010). RGB video annotation systems are proposed in (Yuen et al., 2009) (Vondrick & Ramanan, 2011) (Vondrick et al., 2013) (Ni et al., 2013) (Liu et al., 2008). RGBD video annotation system is proposed in (Teichman et al., 2013). Among these different data types, the RGB video data are easy to access in our daily life due to the low price of cameras.

Segmenting and tracking an object in RGB video is not an easy task. Automatically detail contour annotation is difficult because the object shape changes as it moves and color intensities may change. There are two approaches to label data. One approach is to hand draw the object contour for each frame, but this is time-consuming especially for objects that have complicated shapes. Another approach is to use advance algorithms that can automatically infer the contours based on tracking or learning shape basis.

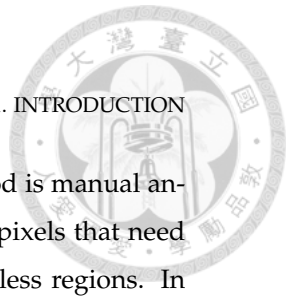
To label detail contours on individual objects in a video, the user is still required to label them one by one at a time in each frame. In order to reduce the effort needed by hand-labeling annotation, the proposed method is to exploit the temporal information in the video to make it an easier task. In the proposed system, the user is requested to label



the first frame of the video with GrowCut (Vezhnevets & Konushin, 2005) in a way that putting seed pixels with different labels to indicate different objects. In each iteration, each seed pixel tries to change the label of its neighbor pixel. If the strength of the seed pixel is higher, the label of its neighbor pixel will be changed to the the same label as the seed pixel. After several iterations, the whole frame is segmented.

In order to assist user labeling, two steps are proposed to find the corresponding seed pixels in the next frame. In the first step, the labels from GrowCut result are propagated to the next frame based on the optical flow result (Farneback, 2003). In the second step, since propagated seed pixels in the next frame may take the wrong labels especially near the edges due to false correspondence given by the optical flow, the labels of the pixels near the edges in the next frame are cleared (set as unlabeled). Since GrowCut performs well on the region that color changes dramatically, the unlabeled pixels near the edges would be assigned labels by applying GrowCut again so that the next frame is segmented. This procedure could be repeated until the desired object contour across the whole video frames are labeled.

Given that the detail object contours are labeled across video frames, they could be used to extract dense pixel-to-pixel correspondences inside the contours. Finding dense pixel-to-pixel correspondences is an important task in both the robotics and computer vision literature. Many tasks including simultaneous localization, mapping and moving object tracking (SLAMMOT) (Wang et al., 2007) and non-rigid structure from motion (Bregler et al., 2000) (Akhter et al., 2008) (Dai et al., 2014) requires correct dense correspondences to improve or evaluate the performance. Dense and correct correspondences help the target tasks be more robust and precise compared to only sparse correspondences since there are more que to accomplish the tasks. For example, tracking the position of a car in a video sequence with dense correspondences is better than using spare correspondences. The reason is that if some of the correspondences are wrongly estimated, we have more chances to recover them by applying dense correspondences since there are more correct correspondences. However, dense correspondences ground truth is hard to obtain, one solution is using synthetic data (Baker et al., 2011) (Butler et al., 2012). A framework that could obtain optical flow ground truth data is proposed in (Baker et al., 2011). It relies on hidden fluorescent texture and high resolution images with very small movement of the scene per frame to enable accurate tracking. Although the ground truth could be obtained by this



method, it is not feasible to apply to our everyday videos. Another method is manual annotation which is very painful to the annotator since there are too many pixels that need to match and many of them are full of ambiguities especially on textureless regions. In order to tackle this problem, a dense correspondence annotation scheme is also proposed in this thesis. We are not going to provide a ground truth; instead, we are going to make our everyday videos a benchmark for evaluating dense correspondence estimation.

Given the detail object contours labeled in the previous step, we are going to find the salient feature correspondences inside the contours between two consecutive frames. Since edges make the user easier to observe the salient color changes inside the object contours and is precise enough to outperform hand-labeling, the Canny edge detector (Canny, 1986) is performed to find the edges inside the contours. The user could easily adjust the threshold of the detector by a scrollbar to find the most feasible edge detection result. A fast optical flow algorithm (Farneback, 2003) is used to find the corresponding points of the edges in the next frame. The user can check if the corresponding points are correctly matched by observing the color of the matched points. If the correspondences found by the optical flow are wrong, the user could clear the wrongly matched points by an eraser and specify the correct correspondences by selecting corresponding edges and drawing corresponding line segments or curves in the two frames. The user could help to find out the correct corresponding points found by the optical flow and is still able to specify the salient feature that the optical flow algorithm and the edge detector could not find out.

Given the edges and the line segments specified by the user, a robust point matching algorithm (Chui & Rangarajan, 2003) is applied. The algorithm could estimate the non-rigid transformation of the two point sets, which is called thin-plate spline (TPS) warping (Bookstein, 1989) (Wahba, 1990). We assume that the salient features specified by the user could guide the warping function to find a smooth correspondences in the textureless region. Abruptly change of correspondences due to self occlusion might lower the performance of the warping function.

Results on labeling different cases of data compared with state-of-the-art systems are shown in Chapter 5. Our proposed system performs well to label detail shape and dense correspondences of objects in a video. The overview of our proposed approach is shown in Figure 1.1

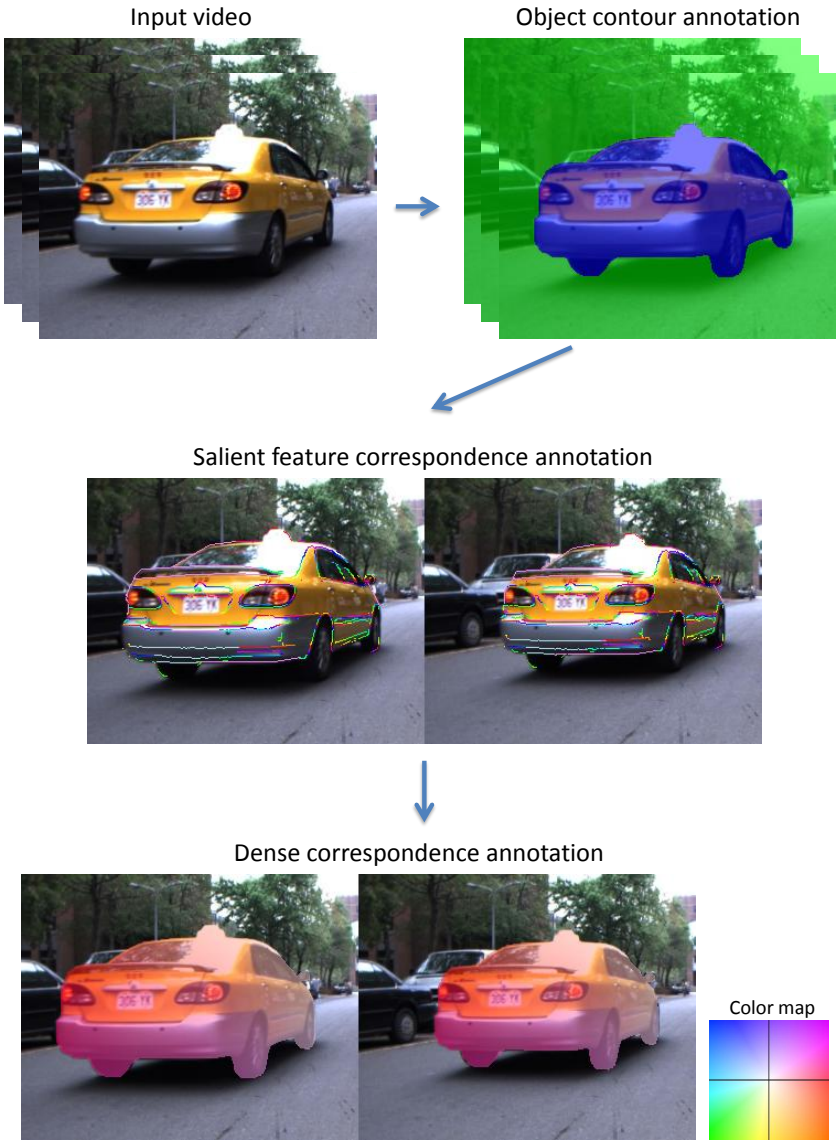
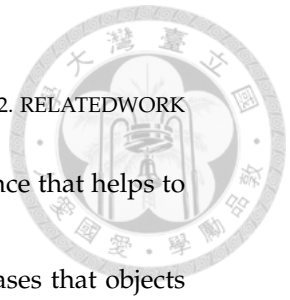


Figure 1.1. Overview of our annotation system

CHAPTER 2

Related Work

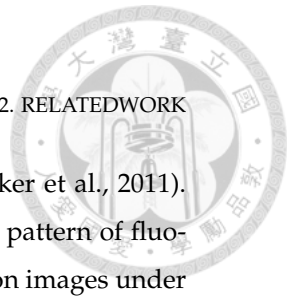
ANNOTATION systems are developed in order to reduce the user's efforts on labeling different kinds of data such as laser range points, RGB, or RGBD data. An annotation system based on laser scanner data and the corresponding image data is proposed in (Weng et al., 2012). They believe that it is more efficient to label the laser scanner data with the corresponding image data. An online image annotation system that not only provides a tool for image annotation but also builds a large image dataset at the same time is proposed in LabelMe (Russell et al., 2008). An annotation system based on video annotation with labeling bounding boxes on key frames is proposed in Vatic (Vondrick et al., 2013). Interpolation and tracking algorithms are performed on the key frames to propagate the labels. However, detail contour annotation is not addressed in this work. A video annotation system based on the heuristic that objects often move at constant velocity and follow straight trajectories is proposed in LabelMe Video (Yuen et al., 2009). Polygons are used to indicate the shape of the labeled object by the user. Positions of the vertices of the polygon could be adjusted. These vertices are called control points. Filling the missing polygons in between two user labeled key frames with interpolation techniques is their approach to reduce the effort of hand-labeling. A motion annotation tool that focuses on object layer segmentation and optical flow estimation with human intervention is proposed by (Liu et al., 2008). Users only need to label the object contours in the first frame and the system would automatically track the positions of the object contours in the remaining video frames. The user could modify the result of contour tracking if it is not good enough. Once the accurate object contours are obtained, dense optical flow is estimated for each object contour. The user could adjust the



parameters of the optical flow and specify additional sparse correspondence that helps to improve the flow estimation.

Although the approach in LabelMe Video could be efficient in the cases that objects move with constant velocity and follow straight trajectories, some problems still remain. The first problem is that the number of the control points on the polygon should be the same throughout the video frames. If the shape of the object is changed due to occlusion or self-occlusion, the shape of the object might become more complicated. Therefore, the number of the control points should be increased to fit the shape of the object. The second problem is that the user is required to label the control points in a way that the corresponding control points in the continuous video frames should be put in the same order, e.g. counter clockwise or clockwise. The third problem is that LabelMe Video could only deal with rigid or semi-rigid object with constant velocity. In contrast to LabelMe Video, since GrowCut is an interactive segmentation algorithm, the labeled contour is generated by the segmentation result. The user is not required to draw the polygons so that there is no need to change the number of the control points and to label the object with the same order of putting the control points in the proposed object contour annotation framwork. Moreover, the user could easily fit the shape of the object with GrowCut in the proposed approach. Since the proposed system focuses on RGB video annotation, exploiting all the useful information that helps label a video such as tracking and segmentation is the main focus. Annotations of rigid or non-rigid objects with constant or non-constant velocity could be propagated across different frames to minimize the user's effort in this proposed system.

The proposed framework in (Liu et al., 2008) is similar to our work instead that they fully rely on optical flow to find the dense correspondences between two consecutive frames. The user is required to label object contours first and optical flow is performed on individual object, the parameters of the optical flow could be adjust by the user. If the result of the optical flow is not satisfying, the user could label sparse correspondences and select the motion type of the object so that the system could adjust the flow by applying the user specified motion type. A problem is that the optical flow of the non-rigid object could not be well estimated. Non-rigid matching method is adopted to overcome this problem in the proposed work.



An approach to obtain optical flow ground truth is proposed in (Baker et al., 2011). They build a scene that can be moved in a very small step. A fine spatter pattern of fluorescent paint is applied to all surfaces in the scene. A pair of high-resolution images under ambient lighting and UV lighting is repeatedly taken by the computer. In this way the system could obtain a natural image and an image with textures from fluorescent patterns simultaneously. The rich patterns and high resolution image enables accurate tracking, which is used to produce ground truth. However, this approach is limited to the specified environment which is not feasible on the general purpose video data.

The main purpose of this thesis is to construct a system that could obtain dense correspondences across video frames given general purpose video data. Semantic segmentation property is reserved and non-rigid object annotation could be tackle with. The problem is that it requires more efforts to label the dense correspondences of an object since exploiting all the human capability to label dense correspondences is our purpose.

CHAPTER 3

Background

GROWCUT (Vezhnevets & Konushin, 2005) interactive segmentation is applied to help the user segment the object in the video while a non-rigid point matching algorithm called thin-plate spline robust point matching (TPS-RPM) (Chui & Rangarajan, 2003) is applied to find out the dense correspondence in the textureless region. GrowCut and TPS-RPM are introduced in this chapter.

3.1. GrowCut interactive segmentation

An interactive segmentation algorithm is a process of assisting the user to cut out the desired object in an image. The user is requested to provide some sparse or dense labels of the objects so that the algorithm could segment the desired object given the labels. GrowCut (Vezhnevets & Konushin, 2005) is one of the interactive segmentation algorithms which is easy to implement yet gives reasonable result and is capable of segmenting multiple objects simultaneously. It is used to help the user segment out the desired object in this work.

GrowCut relies on a cellular automaton (Neumann, 1966) to do segmentation. Given a small number of user-labelled pixels, the rest of the image is segmented automatically by the cellular automaton. Each image pixel is treated as a cell. The automata evolution rule is shown in Algorithm 1.

l_p is the label of the current cell. θ_p is the strength of the current cell where $\theta_p \in [0,1]$ in general. $\overrightarrow{C}_p$ is the feature vector of the current cell. The neighborhood system N is the von Neumann neighborhood with

Algorithm 1 Automata evolution rule

Require: Given initial label l_p^0 and strength θ_p^0 for each cell specified by the user.

for $\forall p \in P$ **do**
  // Copy previous state
  $l_p^{t+1} = l_p^t$
  $\theta_p^{t+1} = \theta_p^t$
  // neighbors try to attack current cell
  for $\forall q \in N(p)$ **do**
    if $g\left(\left\|\overrightarrow{C}_p - \overrightarrow{C}_q\right\|_2\right) \cdot \theta_q^t > \theta_p^t$ **then**
      $l_p^{t+1} = l_q^t$
      $\theta_p^{t+1} = g\left(\left\|\overrightarrow{C}_p - \overrightarrow{C}_q\right\|_2\right) \cdot \theta_q^t$
    end if
  end for
end for

$$N(p) = \{q \in Z^n : \|p-q\|_1 := \sum_{i=1}^{n} |p_i - q_i| = 1\} \tag{3.1}$$

g is a monotonous decreasing function bounded to [0,1],

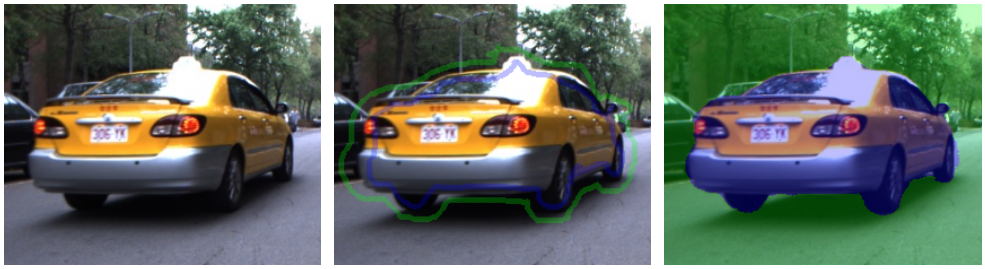
$$g(x) = 1 - \frac{x}{\max \|\overrightarrow{C}\|_2} \tag{3.2}$$

Initial states for $\forall p \in P$ are set to: $l_p = 0, \theta_p = 0, \overrightarrow{C}_p = RGB_p$ where RGB_p is the three dimensional vector of pixel p's color in RGB space. The segmentation is started by specifying initial seed pixels by the user. This is done by the user's stroke with different object labels. For instance, the blue brush stroke indicates the object label while the green brush indicates the background label as shown in 3.1. Once a seed pixel is specified by the user's stroke, its neighbor pixels within a Euclidean distance of 5 pixels are set to the same label with the seed pixel. The strength of the seed pixel is set to 1 while the strength of neighbor pixels are linearly decay to 0 according to their distance to the seed pixel. Based on the initial seed pixel distribution, the whole image is segmented by the automata evolution rule.

The whole image is segmented as the following procedure. Each pixel with none zero strength is tend to make its neighbors the same label with itself. Whether a pixel would change its label is depended on the values of strength and the distance in RGB color space between the pixel and its neighbor pixel. By this rule, every label tend to expand all over

the whole image. The calculation repeated until the automaton converges to a stable condition where the cell states rarely change. If the result of segmentation is not satisfactory, the user could clear the label of the wrongly segmented pixels, put some new seed pixels and perform GrowCut again. The examples of GrowCut interactive segmentation are shown in Figure 3.2 and Figure 3.3. The method is guaranteed to converge as the strength of each cell is increasing and bounded.

Figure 3.1. Example of the GrowCut strokes.



(a) (b) (c)

Figure 3.2. An example of GrowCut. (a) The original image. (b) The image with user-specified seed pixels. (c) The segmentation result generated by the algorithm.

3.2. Theory Of Point Matching

Among many algorithms derived for the point matching problem, iterative closest point (ICP) (Besl & McKay, 1992) is probably the most famous one. ICP is famous for

Figure 3.3. Automata evolution steps. Blue is the object label while green is the background label.

its easy concept while producing reliable matching results given good initial guesses. Although ICP performs well matching two rigid point sets, it is common that the object in the video is deformable due to perspective projection or the object itself is non-rigid. In order to deal with this problem, TPS-RPM (Chui & Rangarajan, 2003) is used to match user-specified corresponding points in the two frames. By matching two user annotated point sets, the transformation function could be used to infer the dense correspondences in the textureless region which is hard to label by the user.

3.2.1. Iterative Closest Point (ICP)

Assume that there are two point sets that one could be transformed to the other through translation and rotation. ICP first find the closest points in the other point set as corresponding points of the original point set. A translation and rotation matrix could be computed by this relationship. The original point set is then transformed to the new position by using this matrix. The whole process is repeated until the result is converge under some criteria such as the summation of the distance of each pair of corresponding points is small than a specified value. ICP estimates the rigid body motion of the point set, which is a very useful technique in matching two rigid point sets.

3.2.2. Thin-Plate Spline Robust Point Matching (TPS-RPM)

The approach of TPS-RPM is similar to ICP; that is, iterative estimating correspondences and transformation of the two point sets. The difference between ICP and TPS-RPM is that TPS-RPM use TPS as the warping function to describe the non-rigid transformation between the two point sets. Moreover, instead of estimating a binary correspondence, TPS-RPM adopts a concept called fuzzy correspondence. That is, a point might be matched to many points in the other point set simultaneously. Under a process of simulated annealing, TPS-RPM iteratively estimates correspondences and transformation of the two point

sets and the fuzzy correspondence would gradually converge to binary correspondence. It is argued that the fuzzy correspondence could bring better performance from the view of optimization. Generally, TPS-RPM aims to optimize the following energy function,

$$E(M,f) = \sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\|x_i - f(v_a)\|^2 + \lambda\|Lf\|^2 + T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai} - \zeta\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}$$
$$subject\ to \sum_{i=1}^{N+1} m_{ai} = 1, i \in \{1,...,N\}\ and \sum_{a=1}^{K+1} m_{ai} = 1, a \in \{1,...,K\}\ with\ m_{ai} \in [0,1] \tag{3.3}$$

The matrix M is the fuzzy correspondence matrix where the inner $N \times K$ part of M defines the correspondences and the extra $N+1^{th}$ row and $K+1^{th}$ column are used to handle the outliers. The second term is the constraint on the transformation where f is the TPS warping function. The third term controls the fuzziness of the correspondence matrix M. By incorporating the concept of annealing, when the temperature T gradually reduced to zero, M becomes binary. T also controls the convergence of the whole process. T is gradually reduced to some final temperature T_{final} by the linear equation, $T^{new} = T^{old} \cdot r$ where r is the annealing rate. The fourth term prevents rejection of too many points as outliers. λ and ζ are the weighting parameters for the corresponding terms. The method for optimizing this energy function is discussed in the following paragraph.

3.2.2.1. Update the Correspondences. For the points $a = 1,2,...,K$ and $i = 1,2,...,N$,

$$m_{ai} = \frac{1}{T} e^{-\frac{(x_i - f(v_a))^T (x_i - f(v_a))}{2T}} \tag{3.4}$$

and for the outlier entries $a = K+1$ and $i = 1,2,...N$,

$$m_{K+1,i} = \frac{1}{T_0} e^{-\frac{(x_i - v_{K+1})^T (x_i - v_{K+1})}{2T_0}} \tag{3.5}$$

and for the outlier entries $a = 1,2,..,K$ and $i = N+1$,

$$m_{a,N+1} = \frac{1}{T_0} e^{-\frac{(x_{N+1} - f(v_a))^T (x_{N+1} - f(v_a))}{2T_0}} \tag{3.6}$$

where v_{K+1} and x_{N+1} are the outlier cluster centers. These equations all incorporate the property that if the distance of the matched points is larger, the value of m is smaller, which means lower confidence of correspondence. Moreover, one could prove that m approaches zero when T approaches zero.

Iterated row and column normalization are performed to satisfy the constraints until convergence is reached,

$$m_{ai} = \frac{m_{ai}}{\sum_{b=1}^{K+1} m_{bi}}, i = 1,2,...,N, \tag{3.7}$$

$$m_{ai} = \frac{m_{ai}}{\sum_{j=1}^{N+1} m_{aj}}, a = 1,2,...K. \tag{3.8}$$

3.2.2.2. Update the Transformation. After dropping the term independent of f, Equation 3.3 becomes,

$$\min_f E(f) = \min_f \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \|x_i - f(v_a)\|^2 + \lambda T \|Lf\|^2 \tag{3.9}$$

where f is the TPS warping function.

3.2.2.3. Thin-Plate Spline (TPS). The warping function between two corresponding point sets y_a and v_a is found by minimizing the following energy function:

$$E_{TPS}(f) = \sum_{a=1}^{K} \|y_a - f(v_a)\|^2 + \lambda \int\int \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right] dxdy \tag{3.10}$$

The points are in 2D since only points on the image plane are discussed in this proposed work. Homogeneous coordinates are used for the point set where one point y_a is represented as $(1, y_{ax}, y_{ay})$. Matrices d and w could be found by an unique minimizer f with a fixed regularization parameter λ.

$$f(v_a, d, w) = v_a \cdot d + \phi(v_a) \cdot w \tag{3.11}$$

where d is a 3×3 affine transformation matrix and w is a $K \times 3$ warping coefficient matrix representing the non-affine deformation. $\phi(v_a)$ is a $1 \times K$ vector for each point v_a where each entry $\phi_b(v_a) = \|v_b - v_a\|^2 \log \|v_b - v_a\|$. $\phi(v_a)$ is related to the TPS kernel which contains the information about the internal structural relationships of the point set. Combining $\phi(v_a)$ with the warping coefficient matrix w generates a non-rigid warping. An example of TPS-RPM is shown in Figure 3.4.

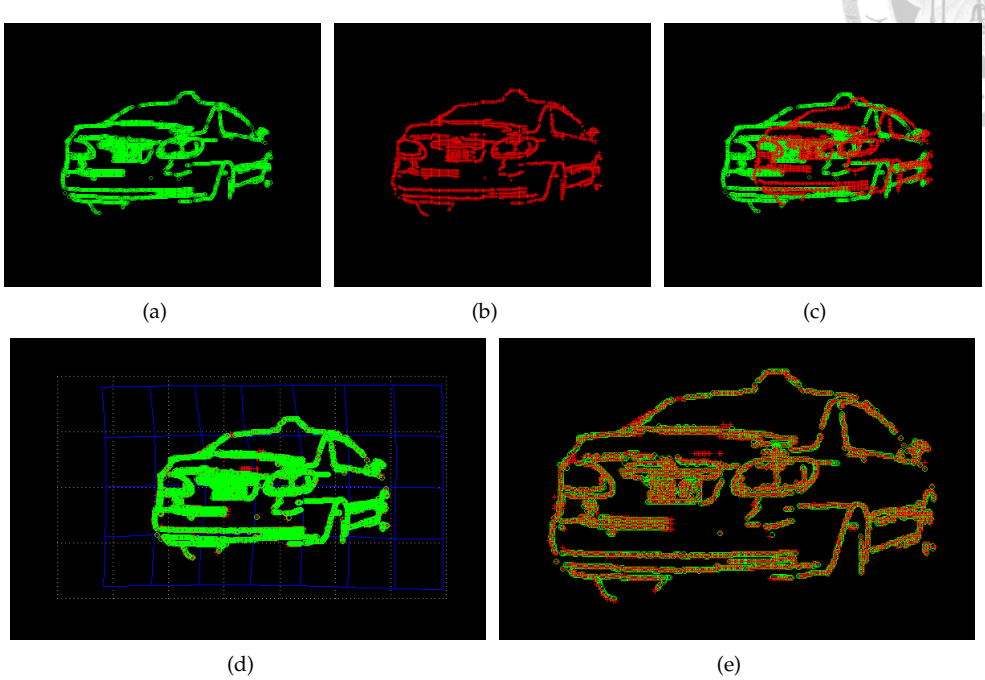


Figure 3.4. (a) The input point set. (b) The target point set. (c) Initial position of the input point set and the target point set. (d) Matching result. The dotted grid indicates the original grid mesh while the blue grid indicates the transformed grid mesh. (e) Matching result without the grids. Some points are regarded as outliers.

CHAPTER 4

Dense Correspondence Annotation

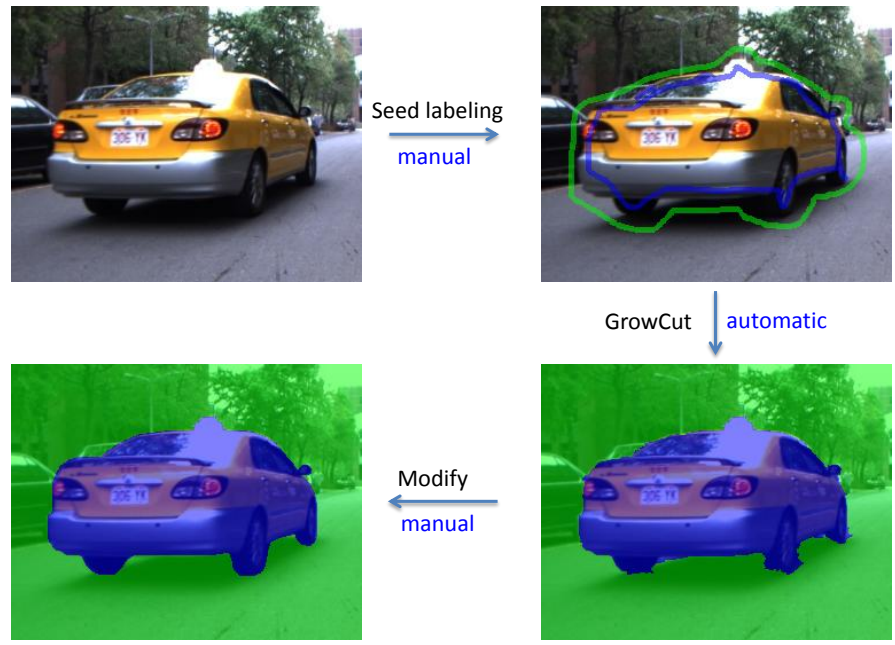
THIS chapter describes the proposed dense correspondence annotation system in detail. The whole framework could be primarily broken into two parts; that is, object contour annotation and dense correspondence annotation inside the corresponding object contours. Object contour annotation is achieved by using an interactive segmentation algorithm and some matching algorithms between two frames to help user labeling. Given the corresponding object contours, information from optical flow and edges are utilized to help user labeling, the user is only required to label the corresponding salient features such as the edges of the object. The correspondences in the textureless region are found by a non-rigid transformation estimated by a non-rigid registration algorithm based on the salient features.

4.1. Object Contour Annotation

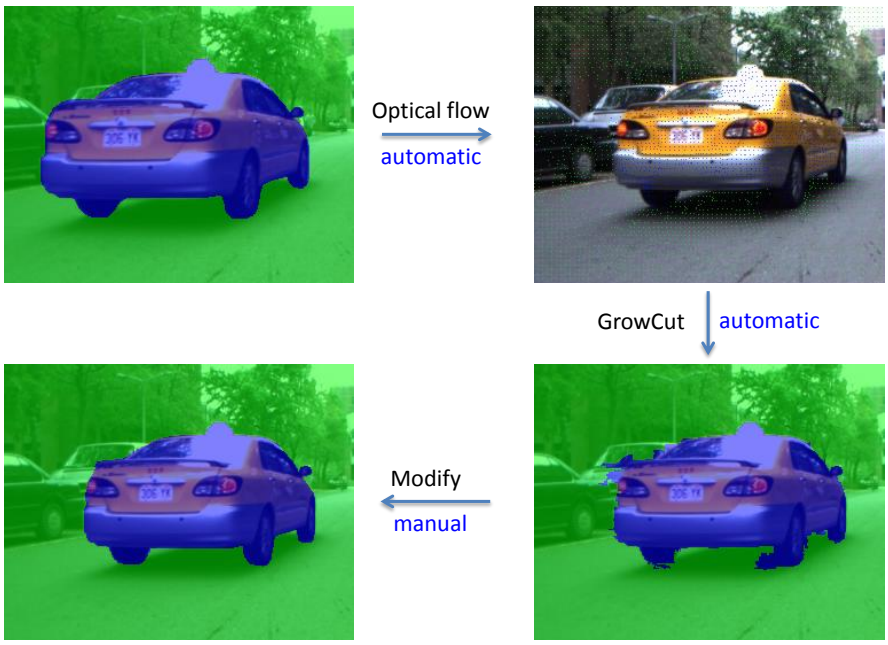
This section discusses the proposed approach of the object contour annotation. In the first step, the user is requested to label seed pixels which represent different objects in the first frame. Based on these seed pixels, the initial segmentation is extracted by an interactive segmentation algorithm - GrowCut. Each segmentation indicates a label of an object. The labels are propagated to the next frame by finding the corresponding labels through an optical flow algorithm (Farneback, 2003). The whole procedure is shown in Figure 4.1.

4.1.1. GrowCut segmentation

A seed pixel is used to define the object label in the nearby region. Once a seed pixel is specified by the user's stroke, its neighbor pixels within a distance are set to the same label



(a)



(b)

Figure 4.1. An overview of the object contour annotation procedure. Blue mask indicates the label of the object while the green one indicates the background. (a) GrowCut segmentation (b) Seed propagation based on optical flow. The next frame is segmented and the labels are modified.

with the seed pixel. The strength of the seed pixel is set to 1 while the strength of neighbor pixels are linearly decay to 0 according to their distance to the seed pixel. The strength of a pixel represents the confidence of the pixel belonging to its label. The strength ranges from 0 to 1 where 1 indicates full confidence. Based on this distribution of the strengths of the pixels, the pixels begin to occupy their neighboring pixels as described in Section . An example of segmenting an object by GrowCut is shown in Figure 4.2. If the result is not satisfactory, there are two ways to modify the contour. One is that the user could clear the labels near the improperly segmented region and put some new seed pixels or not in the cleared region so that GrowCut could be applied again to find the correct contour. The other one is that the user could directly specify the contour of the object by hand labeling. Examples of user modification are shown in Figure 4.3 and 4.4.

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

Figure 4.2. An example of segmenting object by GrowCut. (a) Two labels are specified. (b) Segment by GrowCut. GrowCut performs bad on the the woman's right hand. (c) Clear the labels near the desired contour. (d) Segment again and still some errors remain. (e) Directly specify new labels on the wrongly segmented regions and the woman is segmented.

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

Figure 4.3. An example of clearing the improperly segmented region and apply GrowCut again. (a) The original image. (b) Improperly segmented image. (c) Clear the labels near the desired contour. (d) Apply GrowCut again.

(a) (b) (c)

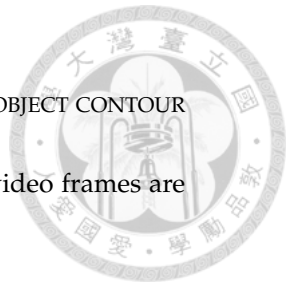
Figure 4.4. An example of directly specify the contour of the object by hand labeling. (a) The original image. (b) Improperly segmented image. (c) Directly specify the contour by hand labeling

4.1.2. Seed propagation

Given the labeled current frame, to propagate labels to the next frame means to segment and label the next frame such that the same labeled region in the current frame and the next frame corresponds to the same object, which is defined by the user. Initial seed pixels are required to segment and label the current frame. These seed pixels can be classified into two categories, labeled seed pixels and null seed pixels. A labeled seed pixel is the seed pixel with label and strength as described in the previous subsection. Null seed pixel is the seed pixel with no label and zero strength. When a null seed pixel is put, it is equivalent to locally resetting all the strength and labels in a neighboring region.

Given the labeled current frame, the corresponding seed pixels in the next frame is found by optical flow. If the edges extracted by edge detector such as Canny detector (Canny, 1986) overlap with label boundaries, the pixels near the edges would be reset to no label since the edges and label boundaries is probably a part of the object contour, resetting the pixels near the edges could also reset the improperly propagated pixel labels from the previous frame by matching process. Given this distribution of seed pixels, GrowCut is applied to segment the next frame. If the user is not satisfied with the result of segmentation, the user could manually input labeled seed pixels. This helps the system to correct and refine the labeled video frame. Except for adding labeled seed pixels, the user can also put null seed pixels so that GrowCut would automatically decide which label the null seed pixels should be.

After initial seed pixels are set, the system segments and labels the whole frame. As long as the seed pixels are appropriately labeled in the first frame and the optical flow algorithm could find out the correct correspondences in the future frames, seed propagation



could keep continuing without user intervention until all the remaining video frames are labeled.

4.2. Dense Correspondence Inside the Object Contour

Given the detail object contour obtained from the previous section, it could be used to infer the dense correspondence of the object across video frames. The user is only required to check the salient feature correspondences due to the ambiguities in the textureless regions. Canny edge detection is used to find out the salient features in the current frame. More powerful features such as SIFT (Lowe, 2004) is not adopted since our observation shows that hand-labeling salient features is almost the same as to find out the edges in the object contour. Optical flow is used to find the corresponding salient features in the next frame. The user could observe the result of matching and adjust the extracted correspondences by clearing false correspondences and labeling correct correspondences. Finally, a non-rigid registration method is adopted to find the correspondences in the textureless region using the adjusted salient feature correspondences.

4.2.1. Salient Feature Correspondences

An intuitive idea of labeling salient feature correspondences is requiring the user to directly observe the consecutive two video frames and carefully specify corresponding salient features by drawing line segments or curves on the two frames as shown in Figure 4.5. However, it requires a lot of time and effort to accomplish this task. In order to tackle this problem, a more efficient annotation procedure is designed. First of all, we observed that intuitively labeling salient features correspondences is a process like labeling the edges so that Canny edge detection is used to find the salient features inside the object contour in the current frame. Since different parameters would heavily affect the result of edge detection, the user could easily adjust the parameter by a scroll bar and directly observe the effect. Optical flow is used to find the corresponding salient features in the next frame, corresponding points are denoted as the same color. If optical flow returns a wrong correspondence, the user could clear the wrong correspondence through the mouse and specify new correspondences by directly drawing on the video frames if needed. If specifying new correspondences is too tedious due to large range false correspondences given by optical flow, the user could specify the corresponding edges instead of directly specifying correspondences by drawing. However, the detail point to point correspondences of

the corresponding edges and the corresponding line segments or curves drawn by the user are unknown. To tackle this problem, a non-rigid registration method is adopted. Different kinds of salient feature correspondences annotation are shown in Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9.

Figure 4.5. An intuitive idea of labeling salient feature correspondences. Corresponding line segments are denoted as the same color.

Figure 4.6. User specified points on the edge that are associated by optical flow. The corresponding points are denoted as the same color

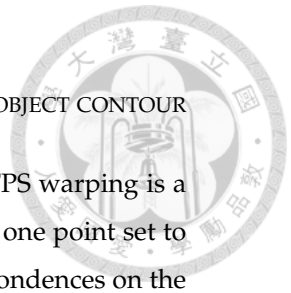
4.2.2. Non-Rigid Registration

Once the salient feature correspondences are obtained, TPS-RPM (Chui & Rangarajan, 2003) is adopted. This method iteratively estimate the correspondence of two point sets

Figure 4.7. User-specified edges

Figure 4.8. User-specified line segments and curves

Figure 4.9. Merge of Figure 4.6, Figure 4.7 and Figure 4.8



and calculate the transformation using thin-plate spline (TPS) warping. TPS warping is a warping function that aims to find the most smoothing transformation of one point set to the other point set. This property makes it suitable for finding the correspondences on the textureless regions.

TPS-RPM aims to optimize the energy function as discussed in Equation 3.3. The proposed annotation framework requires the user to label salient feature correspondences from the result of the optical flow, edges and directly drawing, only the correspondences from the optical flow are known. In order to incorporate the correspondences from optical flow, assumed that the i^{th} point in a point set corresponds to the j^{th} point in the other point set, the element at i^{th} row and j^{th} column of M is assigned to 1 and the other elements at the same row or the same column would be set to 0. In this way, the function is guided to match the already known correspondences. After the algorithm converges, the output warping function is used to find the correspondences in the textureless regions. If there is any pixel in one frame not matched by any pixel in the other frame due to different numbers of pixels in the object contours, interpolation is adopted to find the corresponding pixel. That is, searching for the nearest pixels with correspondences and interpolating the positions of the corresponding pixels in the other frame to find the corresponding pixels (might be in subpixel accuracy) of the unmatched pixels.

CHAPTER 5

Experiment Result

In this chapter, the experiment result of the proposed approach is analyzed. The user interface is also discussed. Optical flow ground truth provided in (Baker et al., 2011) is used to evaluate the system performance. Several cases of real data are labeled using the system proposed in (Liu et al., 2008) and our proposed approach. The result of an optical flow algorithm is also adopted as a comparison. The comparison of the three approaches shows that our method performs best on finding dense correspondences of non-rigid objects.

5.1. User Interface

The user interface is shown in Figure 5.1. Similar to other video annotation systems (Yuen et al., 2009) (Liu et al., 2008), a scrollbar is attached and is capable of adjusting the position of the video. Since our assumption is that dense correspondences should be observable in the consecutive two frames simultaneously, two consecutive video frames are displayed at the same time so that the user could directly observe the differences between two video frames and directly specify correspondences on them. In some cases correspondences could be easier to observe by moving the video frame forward and backward, which could be done by dragging the scrollbar. There are other two smaller scrollbars attached below the video frames that are used to adjust the parameter of edge detection.

An user interface is designed for the dense correspondence annotation. In order to help the user to simultaneously adjust the position of the video frame and compare the correspondences between two video frames, two consecutive video frames are shown at the same time and a scroll bar is attached in the interface to adjust the position of the video frame.

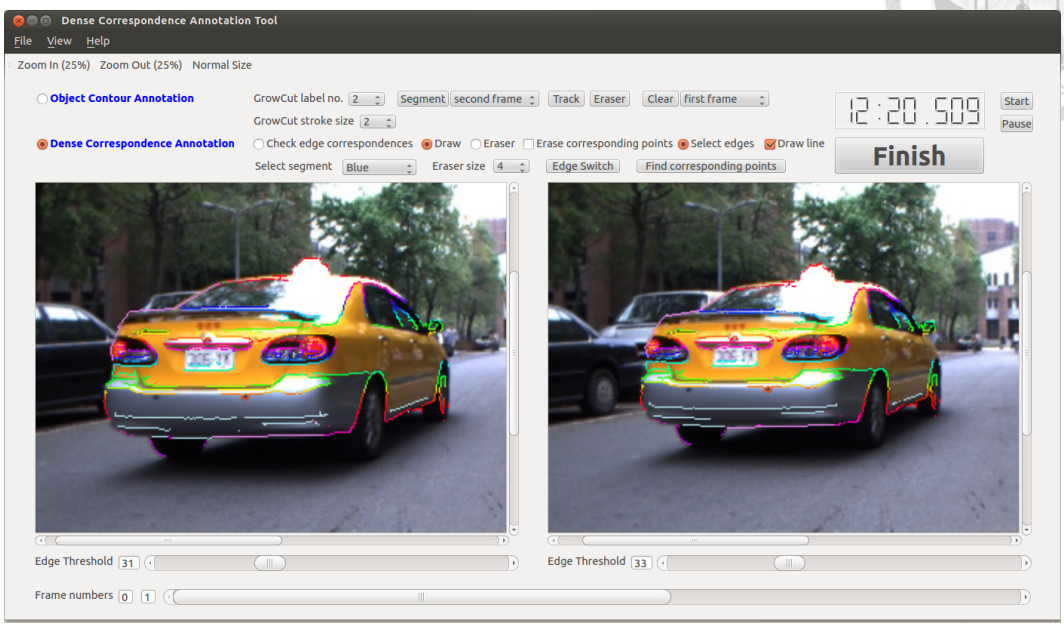


Figure 5.1. A snapshot of our user interface.

5.2. Dense Correspondence estimation on Middlebury Dataset

5.2.1. Error Metric

The most commonly used evaluation metrics for the optical flow algorithm are average angular error (AAE) and average endpoint error (AEE) (Baker et al., 2011). The angular error (AE) between an estimated flow vector (u,v) and the ground truth flow vector (u_{GT},v_{GT}) is the same as the angular error in 3D space between $(u,v,1.0)$ and $(u_{GT},v_{GT},1.0)$. The AE is intuitively defined as,

$$AE = \cos^{-1}\left(\frac{1.0+u\times u_{GT}+v\times v_{GT}}{\sqrt{1.0+u^2+v^2}\sqrt{1.0+u_{GT}^2+v_{GT}^2}}\right) \tag{5.1}$$

Extending the flow vector to 3D space prevent the case of dividing by zero when a pixel is static between two consecutive frames. The endpoint error (EE) between a flow vector (u,v) and the ground truth flow (u_{GT},v_{GT}) is the difference in the length which is defined as,

$$EE = \sqrt{(u-u_{GT})^2+(v-v_{GT})^2} \tag{5.2}$$

5.2.2. Evaluation on the Middlebury Benchmark

Using the error metric defined in the last subsection, the accuracy of flow estimation is measured on the data set shown in Figure 5.2

(a) (b)

Figure 5.2. RubberWhale dataset in the Middlebury benchmark. (a) First frame. (b) Second frame.

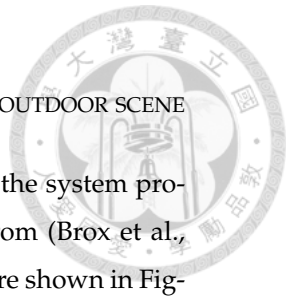
The result of flow estimation of the yellow box in the right and bottom side of the frames is shown in Figure 5.3. The error between our annotation and the ground truth flow is 3.3781° in AAE and 0.1193 pixel in AEE. The error metrics are only computed on the non-black region of the ground truth since the black region indicates that the flow is unmeasurable due to occlusion or noise.

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

Figure 5.3. (a) The first frame. (b) The ground truth flow. (c) The annotated dense correspondences transferred to the flow color code. (d) Difference of (a) and (b). (e) Color map

5.3. Dense Correspondence Estimation on the outdoor scene

The middlebury dataset contains only indoor scenes or synthetic data and the motion between two frames are relatively simple compared to the outdoor scene. Since estimating dense correspondence in outdoor scenes is essential, test data of outdoor scenes is adopted.



In order to test our system performance, annotation results generated by the system proposed in (Liu et al., 2008) and the pure optical flow estimation results from (Brox et al., 2004) (Bruhn et al., 2005) are also presented for a comparison. The results are shown in Figure 5.4, Figure 5.5 and Figure 5.6. The system proposed in (Liu et al., 2008) will be called motion annotation system in the following paragraph.

In Figure 5.4, a taxi is driving away from the camera. The result of (Liu et al., 2008) and our result and capture the different motions of different parts of the taxi and still maintain a smooth field. The result of optical flow generally captures a reasonable flow distribution but is not smooth enough. In this simple case, our method and the method of (Liu et al., 2008) perform relatively better than pure optical flow estimation.

In Figure 5.5, a woman with an umbrella is walking. Compared to the previous case, this case is harder since the camera is moving as well and the shape of the woman changes non-rigidly. With the help of object contour annotation and salient feature correspondence annotation, our approach obtains a smoother flow field compared to the pure optical flow estimation while still captures the detail correspondence of each part of the object compared to the result of (Liu et al., 2008).

In Figure 5.6, a man is riding a bicycle away from the camera. The adopted optical flow algorithm in our framework fails to track the motion of the bicycle so that the correspondences are all labeled by edges or specified by the user. By carefully labeling the rider and the bicycle, a detail contour with hollow parts is generated. The motion annotation system does not support labeling hollow parts of the bicycle and could only label the object by a polygon.

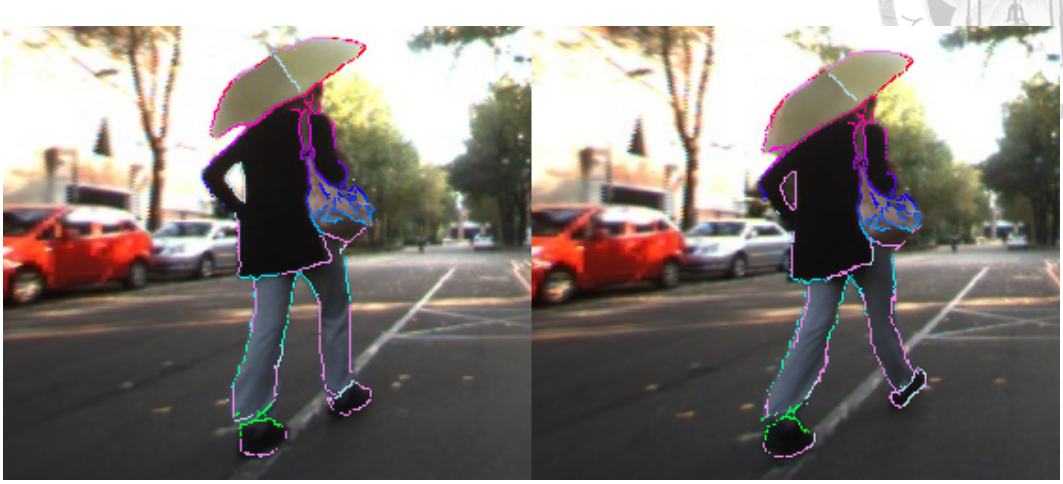
From the three cases, smooth assumption plays an important role of guiding the dense correspondence estimation. The proposed method could obtain a relatively precise estimation result on labeling non-rigid object while maintaining smoothness constraints. However, smooth constraints may violate the actual structure of the object due to self-occlusion such as the man's right leg in the bicycle dataset. This problem might be solved by labeling many segments of the object and combine them together, which is left as future work.

(a)

(b) (c) (d)

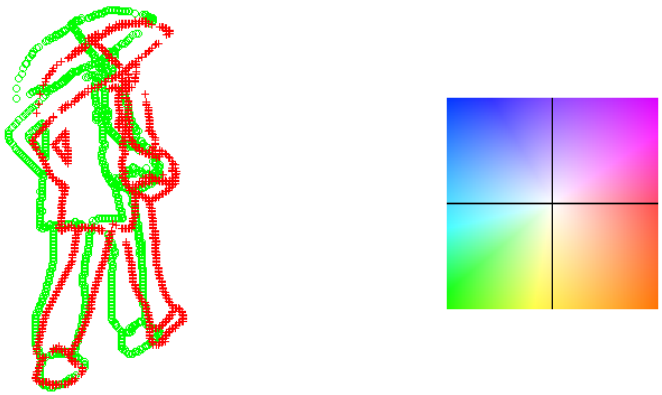
(e) (f)

Figure 5.4. (a) Annotations in two consecutive frames (b) Dense correspondence generated by pure optical flow with an object mask (c) Dense correspondence generated by (Liu et al., 2008) (d) Dense correspondence generated by the proposed method (e) Overlapped annotations in two consecutive frames (f) Color map



(a)

(b) (c) (d)

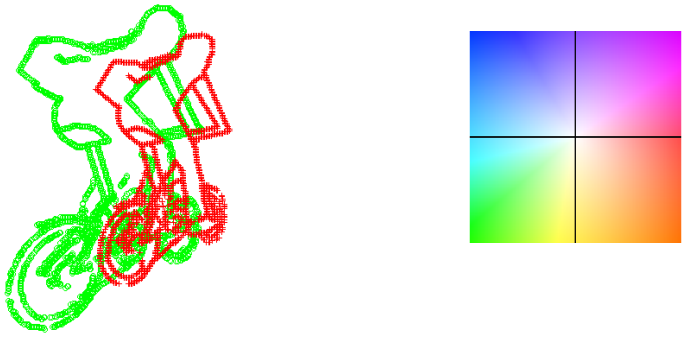


(e) (f)

Figure 5.5. (a) Annotations in two consecutive frames (b) Dense correspondence generated by pure optical flow with an object mask (c) Dense correspondence generated by (Liu et al., 2008) (d) Dense correspondence generated by the proposed method (e) Overlapped annotations in two consecutive frames (f) Color map

(a)

(b) (c) (d)



(e) (f)

Figure 5.6. (a) Annotations in two consecutive frames (b) Dense correspondence generated by pure optical flow with an object mask (c) Dense correspondence generated by (Liu et al., 2008) (d) Dense correspondence generated by the proposed method (e) Overlapped annotations in two consecutive frames (f) Color map

5.4. User study

Eight users are requested to label dense correspondences with the taxi and the woman datasets shown in the previous section using our system. The results of object contour annotation are shown in Figure 5.7, Figure 5.8, Figure 5.9 and Figure 5.10.

(a) (b) (c)

Figure 5.7. (a) The original image. (b) The bluer the pixel means that more user label the pixel as the object. (c) The same as (b) instead that the original colors are not shown.

(a) (b) (c)

Figure 5.8. (a) The original image. (b) The bluer the pixel means that more user label the pixel as the object. (c) The same as (b) instead that the original colors are not shown.

Table 5.1. Correct rates of user labeled contours.

	Taxi frame 1	Taxi frame 2	Woman frame 1	Woman frame 2
User 1	96.9%	97.8%	93.7%	95.2%
User 2	96.6%	96.5%	91.2%	93.7%
User 3	94.9%	95.8%	94.1%	95.2%
User 4	96.3%	96.3%	91.9%	93.8%
User 5	97.0%	97.7%	94.3%	96.0%
User 6	96.7%	97.2%	93.0%	92.9%
User 7	97.1%	98.1%	94.2%	95.5%
User 8	96.3%	96.7%	92.3%	91.9%

(a) (b) (c)

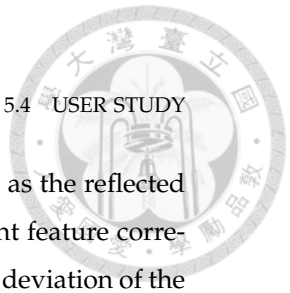
Figure 5.9. (a) The original image. (b) The bluer the pixel means that more user label the pixel as the object. (c) The same as (b) instead that the original colors are not shown.

(a) (b) (c)

Figure 5.10. (a) The original image. (b) The bluer the pixel means that more user label the pixel as the object. (c) The same as (b) instead that the original colors are not shown.

Disagreement of user labeled contours arises on the ambiguous region of the object such as the region that contains similar texture to the background. The users usually guess where the true contour is and label it by the shape prior of the object. Carefully labeled ground truth data is used to check the correctness of the user labeled contour. The correct rates of the user labeled contours are shown in Table 5.1.

When it comes to dense correspondence annotation, the first problem for the user is how to strike a balance between maximizing the desired edges while minimizing noises or



unstable edge through adjusting the edge threshold. Unstable edges such as the reflected tree on the window of the taxi would probably mislead the result of salient feature correspondences annotation since it is not a good feature. The average standard deviation of the taxi dataset is 2.8404 pixel on x coordinate and 0.2606 pixel on y coordinate while that of the woman dataset is 1.0124 pixel on x coordinate and 1.4156 on y coordinate.

CHAPTER 6

Conclusion

Extracting dense correspondences across video frames is important to many tasks in computer vision and robotics. Although state-of-the-art optical flow algorithms could find out a reliable estimation of dense correspondences, there are still many sophisticated motions that the optical flow algorithms can not deal with. The proposed framework requires the user to label all the salient feature correspondences while striking a balance between the effort and the performance. With the help of non-rigid matching algorithm, our method could tackle sophisticated motions that are hard for the optical flow algorithm. However, self-occlusion of the object is still a problem in our approach. Segmenting the object to more small regions and estimate the correspondence with more warping functions might help to solve the problem, which is left as future work.

BIBLIOGRAPHY

Akhter, I., Sheikh, Y., Khan, S., & Kanade, T. (2008). Nonrigid structure from motion in trajectory space. *Advances in Neural Information Processing Systems 21*, 41–48.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, *92*(1), 1–31.

Besl, P. J. & McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256.

Bookstein, F. L. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(6), 567–585.

Bregler, C., Hertzmann, A., & Biermann, H. (2000). Recovering non-rigid 3d shape from image streams. *IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 690–696.

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision*, *3024*, 25–36.

Bruhn, A., Weickert, J., & Schnrr, C. (2005). Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, *61*(3), 211–231.

Butler, D. J., Wulff, J., Stanley, G. B., & Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. *European Conference on Computer Vision*, 611–625.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Chui, H. & Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, *89*(23), 114 – 141.

Dai, Y., Li, H., & He, M. (2014). A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, *107*(2), 101–122.

Farneback, G. (2003). Two-frame motion estimation based on polynomial expansion. *Proceedings of the 13th Scandinavian Conference on Image Analysis*, *2749*, 363–370.

Giro-i Nieto, X., Camps, N., & Marques, F. (2010). Gat: a graphical annotation tool for semantic regions. *Multimedia Tools and Applications*, *46*(2-3), 155–174.

Liu, C., Freeman, W., Adelson, E., & Weiss, Y. (2008). Human-assisted motion annotation. *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Neumann, J. V. (1966). *Theory of Self-Reproducing Automata*. Champaign, IL, USA: University of Illinois Press.

Ni, Y., Poggio, T., Hadjiconstantinou, N. G., & Ni, Y. (2013). Mouse behavior recognition with the wisdom of crowd.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1-3), 157–173.

Teichman, A., Lussier, J., & Thrun, S. (2013). Learning to segment and track in RGBD. *IEEE Transactions on Automation Science and Engineering*, *10*(4), 841–852.

Vezhnevets, V. & Konushin, V. (2005). "GrowCut" - Interactive multi-label n-d image segmentation by cellular automata. In *GraphiCon*.

Vondrick, C., Patterson, D., & Ramanan, D. (2013). Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, *101*(1), 184–204.

Vondrick, C. & Ramanan, D. (2011). Video annotation and tracking with active learning. *Advances in Neural Information Processing Systems 24*, 28–36.

Wahba, G. (1990). Spline models for observational data. *Society for Industrial and Applied Mathematics*.

Wang, C.-C., Thorpe, C., Thrun, S., Hebert, M., & Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, *26*(9), 889–916.

Weng, C.-C., Wang, C.-C., & Healey, J. (2012). A segmentation and data association annotation system for laser-based multi-target tracking evaluation. *IEEE/ASME International*

Conference on Advanced Intelligent Mechatronics, 80–86.

Yuen, J., Russell, B. C., Liu, C., & Torralba, A. (2009). Labelme video: Building a video database with human annotations. *International Conference on Computer Vision*, 1451–1458.

Document Log:

Manuscript Version 1 — 29 September 2014)

Typeset by $\mathcal{AMS}$-LaTeX — 13 November 2014

YEN-TING CHEN

THE ROBOT PERCEPTION AND LEARNING LAB., DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION ENGINEERING, NATIONAL TAIWAN UNIVERSITY, NO.1, SEC. 4, ROOSEVELT RD., DA-AN DISTRICT, TAIPEI CITY, 106, TAIWAN, *Tel.* : (+886) 2-3366-4888 EXT.407

E-mail address: r01922116@ntu.edu.tw