SUPERPIXEL BASED IMAGE SEQUENCE REPRESENTATION AND MOTION ESTIMATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KUTALMIŞ GÖKALP İNCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2017

Approval of the thesis:

SUPERPIXEL BASED IMAGE SEQUENCE REPRESENTATION AND MOTION ESTIMATION

submitted by **KUTALMIŞ GÖKALP İNCE** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. A. Aydın Alatan _____
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Prof. Dr. Mübeccel Demirekler _____
Co-supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. A. Enis Çetin
Electrical and Electronics Engineering Dept., Bilkent University _____

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Osman Serdar Gedik
Computer Engineering Dept., Yıldırım Beyazıt University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: KUTALMIŞ GÖKALP İNCE

Signature :

ABSTRACT

SUPERPIXEL BASED IMAGE SEQUENCE REPRESENTATION AND MOTION ESTIMATION

İnce, Kutalmış Gökalp
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. A. Aydın Alatan
Co-Supervisor : Prof. Dr. Mübeccel Demirekler

January 2017, 96 pages

In this study a superpixel based representation of image sequences is proposed. For superpixel extraction, a novel gradient ascent approach, in which spatial and spectral statistics are utilized to obtain an optimal Bayesian classifier for pixel to superpixel label assignment, is proposed. Utilization of the spectral and spatial statistics reduce the dependency on user selected global parameters, while increasing the robustness and adaptability. Proposed Local Adaptive Superpixels (LASP) approach exploits hexagonal tiling, while achieving some refinement during initialization in order to improve the computation time and accuracy. The experiments conducted on Berkeley segmentation database show that LASP outperforms the existing methods in terms of boundary recall and computation time. Moreover, the proposed method provides lower bleeding error performance compared to the existing gradient ascent techniques. In order to obtain temporally consistent superpixels, a superpixel based occlusion aware layered motion estimation method is also proposed. Proposed motion estimation method combines a Bayesian method with well known gradient descent approaches for optical flow estimation. Proposed method is able to handle occlusions and large displacements. Experiments conducted on Middlebury Database show that performance of the proposed motion estimation method is comparable to state-of-the-art methods, while providing a less computationally complex solution. Using the output of the motion estimation algorithm, the superpixels in the previous

frame placed on the current frame, which provide an initial estimate for superpixels on this frame. Refining this estimate with the information on current frame, it becomes possible to obtain temporally consistent superpixels. These superpixels can be utilized for the representation of image sequences. This representation is developed for video object segmentation, but might also be utilized for various computer vision problems like compression, object tracking and background modeling.

Keywords: Superpixel, Over Segmenation, Motion Estimation, Temporal Superpixel, Mean-Shift, KLT

ÖZ

SÜPER PİKSELLER İLE GÖRÜNTÜ DİZİLERİNİN BETİMLENMESİ VE HAREKET KESTİRİMİ

İnce, Kutalmış Gökalp
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. A. Aydın Alatan
Ortak Tez Yöneticisi : Prof. Dr. Mübeccel Demirekler

Ocak 2017 , 96 sayfa

Bu çalışmada, görüntü dizilerinin süper pikseller ile betimlenmesi için bir yöntem sunulmaktadır. Süper piksel çıkarımı için uzamsal ve spektral istatistiklerden faydalanan yeni bir artan eğim algoritması, piksel - süper piksel atamasında optimal Bayes Sınıflandırıcısına erişmek amacıyla önerilmiştir. Uzamsal ve spektral istatistiklerin kullanımı, süper piksel çıkarımında kullanıcı seçimli global parametrelere bağımlılığı azaltırken, algoritmanın daha gürbüz ve adaptif olmasını sağlamıştır. Önerilen Yerel Adaptif Süper Piksel (YASP) yaklaşımı süper pikselleri altıgenler ile ilklerken, ilkleme sırasında yapılan iyileştirmeler ile algoritmanın doğruluğu ve işlem süresinde iyileşme sağlanmıştır. Berkeley Bölütleme Veri Tabanı üzerinde yapılan deneylerde, YASP yaklaşımının sınır belirleme ve işlem süresi bakımından mevcut yöntemlerden daha başarılı olduğu görülmüştür. Önerilen yöntem, diğer artan eğim algoritmalarına göre de daha düşük bir taşma hatası sağlamaktadır. Bu çalışmada, zamanda tutarlı süper piksellerin oluşturulması için, süper piksel tabanlı bir hareket kestirim algoritması da önerilmiştir. Önerilen hareket kestirim algoritmasında, Bayes yaklaşımını optik akı çözümünde kullanılan azalan eğim yaklaşımı ile bir araya getirilmiştir. Önerilen bu yöntem büyük ötelemeler ve nesne geçişmelerini çözebilmektedir. Middlebury Veri Tabanı üzerinde yapılan deneyler, önerilen yöntemin literatürdeki yöntemler ile benzer doğrulukta sonuçlar üretirken, daha az karmaşık bir çözüm sunduğunu göstermektedir. Önceki karedeki süper piksellerin, elde edilen hareket bilgisi ile gelecek kareye

taşınması ile mevcut karedeki süper pikseller ilklenebilmektedir. Mevcut kare kullanılarak bu ilk kestirimin düzeltilmesi ile zamansal tutarlı süper piksellerin elde edilmesi mümkün olmaktadır. Bu süper pikseller görüntü dizilerinin betimlenmesinde kullanılabilir. Bu betimleme video - nesne bölütlemesi için geliştirilmiş olsa da, sıkıştırma, nesne takibi ve arka plan modelleme gibi pek çok bilgisayarla görü probleminde de kullanılabilir.

Anahtar Kelimeler: Süper-Pixel, Aşırı Bölütleme, Hareket Kestirimi, Zamansal Tutarlı Süper Pixel, Ortalama Kaydırma, KLT

to my father

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my supervisor Prof. A. Aydın Alatan and co-supervisor Prof. Mübeccel Demirekler for their invaluable guidance and constant encouragement.

I would like to thank my colleagues Yoldaş Ataseven, Burak Özkalaycı, Eda Bayram and Seçkin Özsaraç for their support. It was a pleasure for me to discuss various topics on this work with Cevahir Çığla and to get his brilliant ideas. I also thank to Filiz Fidan Kurt for proof reading.

Special thanks to my employer ASELSAN for providing flexible work hours which made possible to complete this work and my administrators Hüseyin Yavuz and Mehmet Karakaş for their encouragement.

Most of all I would like to thank to my parents who gave me the passion of learning and guided me to the academic work.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

SP	Superpixel
SV	Supervoxel
TSP	Temporal Superpixel
TP	Turbo Pixels
SuTP	Speeded-up Turbo Pixels
SEEDs	Superpixels Extracted via Energy-Driven Sampling
SLIC	Simple Linear Iterative Clustering
LASP	Local Adaptive Superpixel
RGB	Red Green Blue
LoG	Laplacian of Gaussian
MAP	Maximum A Posteriori
HS	Horn-Schunck
LK	Lucas-Kanade
KLT	Kanade-Lucas-Tomasi
MS	Mean-Shift
ICM	Iterated Conditional Modes
BP	Belief Propagation
EE	End-Point Error
AE	Angular Error
IE	Interpolation Error
NIE	Normalized Interpolation Error
MSE	Mean Squared Error

NOTATION

I_k	observed frame at time k
x	spatial index
c	image channel index
k	time index
$\mu_{c,i}$	spectral mean of channel c of SP i
$\sigma^2_{c,i}$	spectral variance of channels c of SP i
L_k	label image for frame k
$\Omega_{i,k}$	region of support for SP i at frame k
$\Omega^{(v)}_{i,k}$	visible region of SP i at time $k+1$ on frame k
$\Omega^{(o)}_{i,k}$	occluded region of SP i at time $k+1$ on frame k
R_{ij}	relative layer order of SPs i and j
u_i	motion vector of SP i from frame k to $k+1$
$I_{h,k}$	horizontal derivative image for frame k
$I_{v,k}$	vertical derivative image for frame k
I_t	temporal derivative from frame k to $k-1$
s_{ij}	similarity of SPs i and j
b_{ij}	common boundary length of SPs i and j

CHAPTER 1

INTRODUCTION

Image segmentation is a well-studied problem in computer vision [1, 2, 3, 4]. Video object segmentation is closely related to the image segmentation, but for the video object segmentation problem, multiple images and motion information can be utilized. Solution of the segmentation problem and the optical flow field depends on each other, since for the optical flow solution, occlusion boundaries are needed to allow the discontinuities in the motion field and motion information is a significant data for segmentation. Even if the optical flow problem is also a well-studied problem [5, 6], the researches dealing with the solution of the optical flow and segmentation problems together or optical flow solution based on segmentation are limited [7, 8]. Although there is an extensive literature on tracking and data association problem, [9, 10, 11, 12] which might be also useful for the segmentation and optical flow problems, these methods are not commonly employed for video object segmentation.

In this study, a superpixel-based approach for image sequence representation and motion estimation is proposed, which can be utilized for video object segmentation. For the initial over-segmentation superpixels (SPs) are utilized. For temporal propagation of the segmentation information, an SP based motion estimation method is proposed.

The selection of the spatial smoothness (or convexity) parameter, the number of SPs, and the initial grid of the SPs significantly affect the performance of the SP extraction for gradient ascent based methods. Before going further and trying to solve the problems caused by the errors in SP extraction in higher level processes, it is aimed to improve the performance of the SP extraction. For this purpose, a new SP extraction method named Local Adaptive Super Pixels (LASP) is proposed in Chapter 2. LASP

is defined and compared to the state-of-the-art methods.

The proposed SP extraction method is shown to be a powerful way to represent still images; however, in order to utilize the method in video object segmentation, either the SPs between adjacent frames should be associated or the method should be modified to generate consistent SPs between adjacent frames. SPs provide over-segmented images. The over-segmentation problem is an under-determined problem since there are many SP solutions which may result in the same segmentation output. Therefore it is hard to deal with the ambiguity in the SP matching between adjacent frames. If the proposed method can be combined with a motion estimation method, then the SPs in the previous frame can be utilized as initial estimates for SP extraction in the current frame. In Chapter 3, an SP based motion estimation method is presented. The method is tested on Middlebury Database and utilized for consistent SP extraction.

Having the SPs in one frame and the motion information between adjacent frames, it becomes possible to obtain temporally consistent SPs. A method for extracting temporally consistent SPs is presented in Chapter 4. Like SPs providing a less redundant representation of a single image, temporally consistent SPs provide a less redundant representation of image sequences. Since motion information is available in addition to spatial and spectral information of consistent SPs, they can be utilized in various computer vision applications especially in video object segmentation.

Conclusions and the future work are presented in Chapter 5.

CHAPTER 2

SUPERPIXELS

The purpose of superpixel (SP) extraction is to cluster pixels having similar spatio-spectral characteristics for obtaining an efficient representation of an image [13]. SP extraction is considered as a pre-processing step which provides an over-segmented image for higher level analysis. The main requirements for the SP extraction methods are addressed as local structure preservation, avoidance of under-segmentation, obtaining similarly shaped and sized SPs and low computational complexity [14].

In this chapter, a novel SP extraction method is proposed and compared against the state-of-the-art methods. The chapter is organized as follows: in the first section, gradient ascent based SP extraction methods are introduced. The proposed method is presented in the second section, which is followed by the experiments and conclusions.

2.1 Gradient Ascent Methods

The best performing gradient ascent SP extraction methods are based on three basic approaches: Turbo Pixels (TP) [15] starts from initial clusters and generates SPs via region growing, whereas Speeded-up Turbo Pixels (SuTP) [16] and Simple Linear Iterative Clustering (SLIC) [17, 18] begin from initial clusters obtained from a square grid and refine the clusters iteratively, and finally, Super Pixels Extracted via Energy-Driven Sampling (SEEDS) [19] starts from an initial square grid and generates new clusters by dividing the initial clusters. The main idea behind these basic approaches is to combine the spatial and spectral distances by a weight (called as con-

vexity weight [16] or spatial proximity weight [18]), and decide on the pixel labels by minimization of the following weighted pixel ($\boldsymbol{x}$) to cluster (i) distance ($d_i(\boldsymbol{x})$):

$$d_i(\boldsymbol{x}) = f_1(I(\boldsymbol{x}), I_i) + \lambda f_2(I(\boldsymbol{x}), \boldsymbol{x}_i) \tag{2.1}$$

where I denotes the image, $\boldsymbol{x}$ is the position of the pixel, I_i and $\boldsymbol{x}_i$ are the spectral and spatial distributions of cluster (i), f_1 and f_2 are spectral and spatial penalty functions, respectively, and λ is the convexity weight.

The existing problems with these basic methods can be summarized as follows:

- Performances of SuTP [16] and SLIC [17, 18] depend on the selection of the initial grid, in which squares are exploited extensively. SEEDS [19] is not affected significantly by the selection of the initial grid, but it results in irregular SPs in terms of shape and size.
- Performances of SuTP [16], SLIC [17, 18] and SEEDS [19] significantly depend on the selection of the convexity weight (λ) which is kept constant for the whole image. Hence, the methods result in irregularly shaped SPs in textured regions for the sake of uniformly distributed SPs along textureless regions or vice-versa.
- In SLIC [17, 18], the proposed update rule is applied for each pixel at every iteration; however, updating the non-boundary pixels results in high computational complexity and disconnected regions.

Based on these observations, the first conclusion is that the initial tiling is a crucial step for such iterative methods. It is shown that hexagon honeycomb tiling is the most possible convex shape for equally sized partitions on an infinite plane [20]. An implementation of SLIC with hexagonal initial tiling is also proposed [21]. The proposed method in this study is also designed as a gradient ascent technique, starting from a regular grid and updating the cluster memberships and models through spectral similarity and spatial proximity. However, the proposed method improves state-of-the-art by the following modifications:

- Initial tiling is performed in terms of hexagons (honeycomb) which increases the convexity.

- Initial grid is refined by using pre-defined re-segmentation and gradient information.

- Explicit utilization of estimated spectral and spatial variances results in an optimal Bayesian classifier.

- Update of cluster membership is performed only over the boundary pixels. Moreover, update/no update state of each cluster are determined according to the updates in neighbor clusters.

2.2 Proposed Method

In the proposed algorithm, SP boundary pixels are assigned to a cluster (i.e. superpixel) whose probabilistic model is estimated and updated through iterations. For this purpose, given the cluster conditional distributions, label of a pixel can be obtained by maximizing the following likelihood (2.2), after observing pixel X:

$$L(\boldsymbol{x}) = \arg\max_{i \in X(i)} p(X|i) \tag{2.2}$$

where $X \in R^n$ is a feature vector associated with the current pixel, i is the unknown label of the selected cluster (i.e. superpixel) for X, $X(i)$ is the set of clusters that X can be assigned and L is the output label image. In the proposed algorithm, a pixel can only be assigned to clusters within its immediate neighborhood. Feature vector X might include spectral characteristics (i.e. Lab or RGB values), position, relationship between the neighboring pixels (such as texture), temporal characteristics and even some other appearance or geometric properties. For simplicity and reduced complexity, the feature vector is usually defined as the current observation of the pixel itself (2.3), which is not related to the neighboring pixels or frames:

$$X = [I_1(x,y) \ \ldots \ I_C(x,y) \ x \ y]^T \tag{2.3}$$

where $[I_1(x,y) \ \ldots \ I_C(x,y)]$ is observed visual data over C image channels (i.e. RGB, Lab, etc.) and $[x \ y]$ is the position on the image plane (spatial indices of the image channels will be omitted for notational simplicity in the rest of the Chapter). If the cluster conditional distributions of X are assumed to be multivariate Gaussians,

then the likelihood function is defined as:

$$p(X|i) = \frac{1}{(2\pi)^{n/2}\left|\Sigma_i\right|^{-1/2}} exp\left(-\frac{1}{2}(X-\mu_i)^T\Sigma_i^{-1}(X-\mu_i)\right) \tag{2.4}$$

with different mean vectors μ_i and covariance matrices Σ_i for each cluster i. Taking the logarithm of (2.4), the maximization problem can be expressed as a minimization problem:

$$\arg\min_{i\in X(i)} (X-\mu_i)^T\Sigma_i^{-1}(X-\mu_i) + \ln\left|\Sigma_i\right| \tag{2.5}$$

Spectral and spatial distributions are assumed to be independent by noting that the same color can be observed at distant locations. Similarly, since regular and convex SP shapes are preferred, two spatial axes are also assumed to be independent as well. If the channels are orthogonal (i.e. Lab color space), they can also be assumed to be independent. Hence, the minimization problem is expressed as follows (2.6):

$$\arg\min_{i\in X(i)} \sum_{c=1}^{C} \frac{(I_c-\mu_{c,i})^2}{\sigma_{c,i}^2} + \ln\sigma_{c,i}^2 + \frac{(x-\mu_{x,i})^2}{\sigma_{x,i}^2} + \frac{(y-\mu_{y,i})^2}{\sigma_{y,i}^2} + \ln\sigma_{x,i}^2\sigma_{y,i}^2 \tag{2.6}$$

where C is the number of image channels. At this point, it should be noted that if a priori cluster probabilities are equal, minimization of (2.6) simply corresponds to an optimal minimum error-rate Bayesian classifier (i.e. maximum a posteriori cluster probabilities after observing X) for label assignments. However, since the label assignments must be performed iteratively, cluster conditional distributions (their parameters) are re-estimated during iterations. If an initial cluster is composed of multiple uniform regions that yield an initial (erroneous) high variance for intensities, then it might not be possible to converge to the correct label assignments, since increasing the spectral variance decreases the pixel-cluster distance significantly on the tails of the Gaussian distribution. To avoid such problems, the variances are naively discarded to obtain a cost function [18] as:

$$\sum_{c=1}^{C} (I_c-\mu_{c,i})^2 + \lambda\left((x-\mu_{x,i})^2 + (y-\mu_{y,i})^2\right) \tag{2.7}$$

where λ is a global weight supporting the spatial proximity. In [16] a similar cost function is utilized, but L1 norm is preferred for color distance instead of L2 norm, which simply corresponds to a Laplacian distribution rather than a Gaussian distribution. Obviously, λ should be adjusted with respect to the SP size, since for large SPs

spectral similarity outweighs spatial proximity and vice versa for small SPs. In [18], such an approach is proposed to normalize λ with the average SP area. However, the problem with a global spatial proximity (or convexity) weight cannot be solved by simply varying λ with respect to SP size either. In order to understand this major problem, consider two different regions; one has low contrast and the other has high spectral variance (or high contrast). If a small λ is selected, then in the high variance region, the algorithm will produce irregularly shaped clusters, since spectral distance will outweigh the spatial regularization term. On the other hand, for a larger λ it might be impossible to obtain SPs correctly in low contrast regions. The effect of the λ parameter is demonstrated on Figure 2.1. In (b) the SPs on the dancer's dress are irregular ($\lambda = 0.1$), in (c) around the dancer's arms under-segmentation errors can be observed ($\lambda = 0.4$), in (d) the SPs on the dancer's dress are regular and there is no under-segmentation error around the dancer's arms (λ adaptive). The results obtained

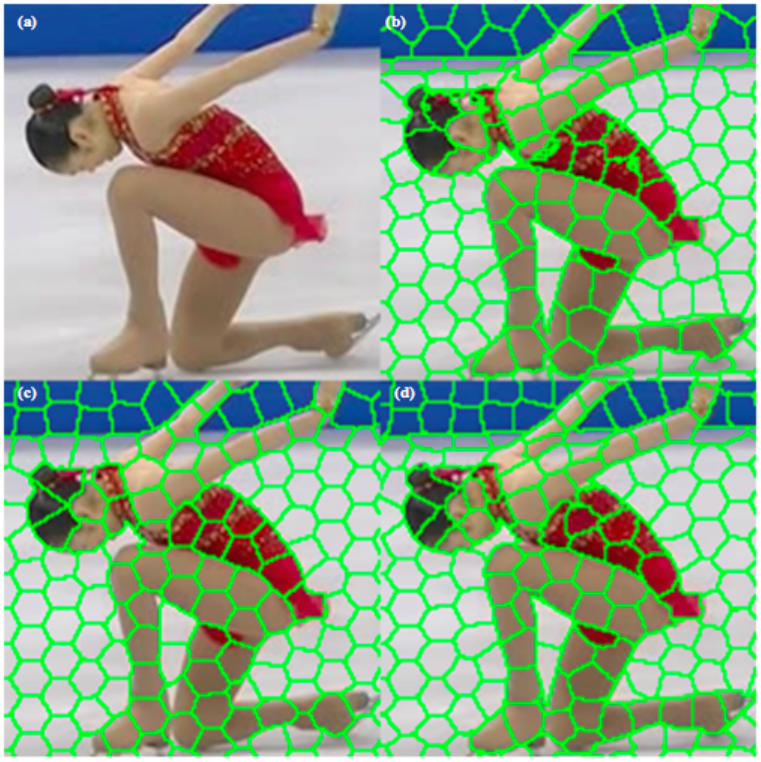

Figure 2.1: Superpixel results for different convexity weights: (a) original image (b) output for $\lambda = 0.1$ (c) output for $\lambda = 0.4$ (d) output for the local adaptive λ.

with proposed and state-of-the art methods are presented in Figure 2.2. Proposed method generates convex shaped and regular SPs whereas SEEDS generates irregularly shaped and sized SPs in general, SuTP and SLIC generate irregularly shaped

SPs on the textured regions. Hence, instead of using a constant λ, variances in (2.6) should be estimated.

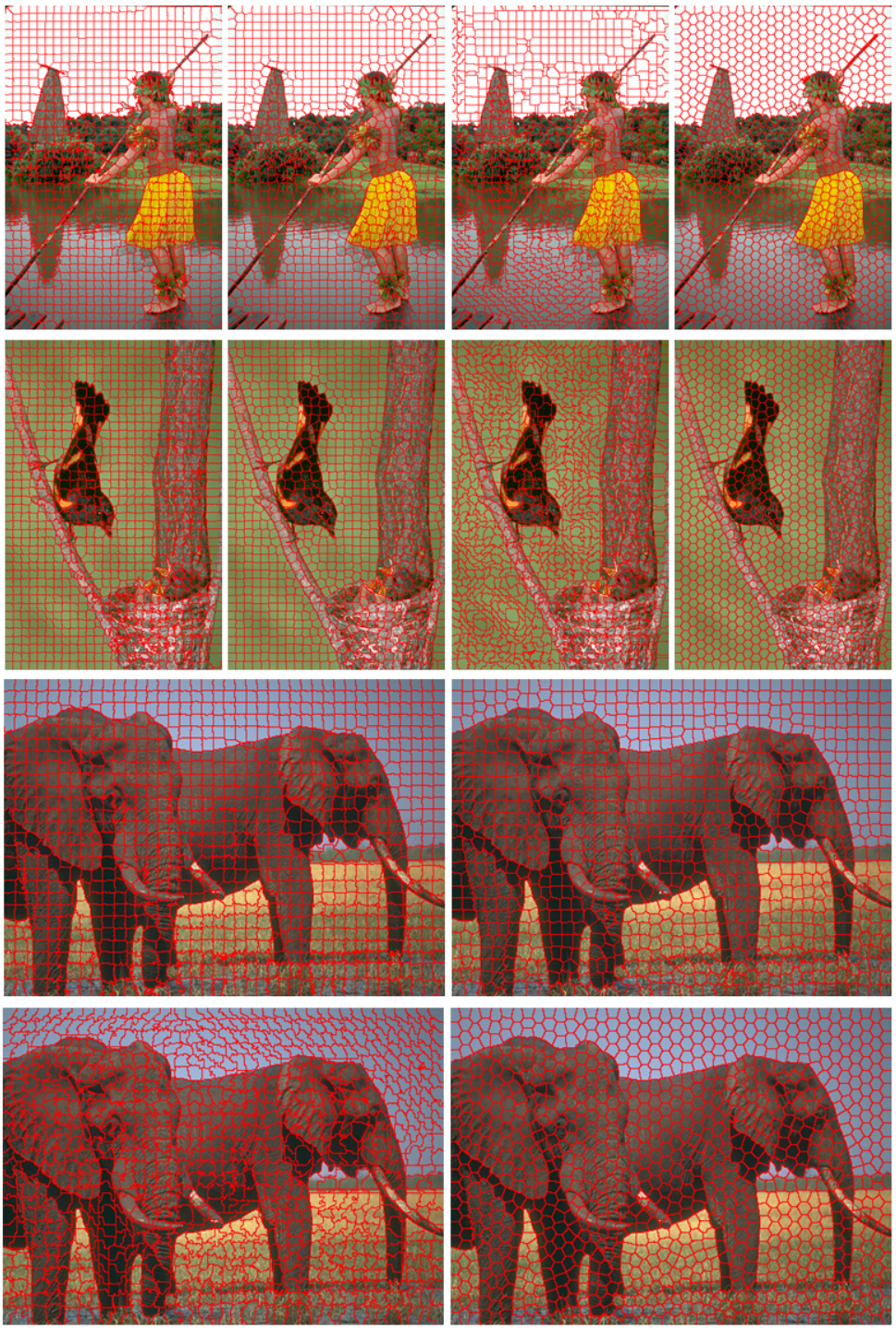

Figure 2.2: Outputs of SuTP [16], SLIC [18], SEEDS [19] and LASP from left to right, respectively.

2.2.1 Local Adaptive Super Pixels

Local Adaptive Super Pixels (LASP) is proposed to extract SPs having regular distribution with convex shapes, as well as similar areas, with high accuracy in low complexity by utilizing finer cluster initialization and normalized cluster to pixel distances. In the ideal scenario, (2.6) should be utilized by the correct (or known) cluster distribution parameters; however, in practice, these parameters must be estimated starting from the initial clusters. First of all, spatial variances, $\sigma_{x,i}^2$ and $\sigma_{y,i}^2$, in (2.6) are assumed to be constant among the candidate clusters, since it is required to have SPs with similar sizes. Similarly, if a pixel has same field of view along horizontal and vertical directions, the horizontal and vertical variances can also be assumed to be same, $\sigma_{x,i}^2 = \sigma_{y,i}^2$ to obtain:

$$\arg\min_{i \in X(i)} \sum_{c=1}^{C} \frac{(I_c - \mu_{c,i})^2}{\sigma_{c,i}^2} + \frac{(x - \mu_{x,i})^2 + (y - \mu_{y,i})^2}{\sigma_{sp}^2} + \sum_{c=1}^{C} \ln \sigma_{c,i}^2 \tag{2.8}$$

where $\sigma_{sp}^2 = A_{sp}/\lambda$ is the spatial variance and A_{sp} is the average SP area. For the spatial variance, global or local average size of the clusters might also be employed. The local average could be defined as the average size of the candidate clusters. On the other hand, utilization of estimated cluster spectral variances, $\sigma_{c,i}^2$ for the normalization of pixel-to-cluster distance might mislead the algorithm during label assignment, since clusters which have multi-modal distributions initially, would like to keep the pixels belonging to different Gaussians, whereas discarding the spectral variance results in the problem of selecting a global spatial proximity (convexity) weight, which is not working well either. To overcome this problem, it is proposed to use a robust estimate for the spectral variance for each pixel by combining the variances of candidate clusters. In order to estimate the robust pixel specific spectral variance, various alternatives, such as the mean, median or minimum variance of the candidates, can be utilized. If the average variance of the candidate clusters is utilized then the minimized cost function turns into the Fisher's criteria [22]. In the proposed algorithm, the minimum spectral variance among the candidate clusters is considered as the robust estimate of the variance for each image channel in (2.9):

$$\arg\min_{i \in X(i)} \sum_{c=1}^{C} \frac{(I_c - \bar{I}_{c,i})^2}{\sigma_c^2(x,y)} + \frac{(x - \bar{x}_i)^2 + (y - \bar{y}_i)^2}{\sigma_{sp}^2} \tag{2.9}$$

where $\sigma^2_{c,i} = \min\left\{\sigma^2_{c,i}\right\}_{i \in X(i)}$. Since the spectral variances of the clusters are required to be small, selecting the minimum variance can be viewed as a good alternative among others.

During iterations the pixel labels are updated with (2.9). Since it is aimed to obtain connected regions, the update rule is applied only to boundary pixels. Such an approach significantly reduces the computation time, as in [16]. Another computational optimization is to stop the iterations for each SP independently. If an SP and its neighbor SPs are not updated during the last iteration, boundary of the corresponding SP cannot be updated at the next iteration, therefore boundary pixels of those clusters are not controlled at the next iteration. This approach helps to increase the number of iterations, which improves the accuracy, without increasing the computation time.

In this study, a regular grid of hexagons (comb) is utilized during the initialization, since they yield more compact representation compared to common square SPs. Once the cluster centers on the image plane are set, each pixel is assigned to the "nearest" cluster center on the image plane and spectral distributions are initiated. However, hexagon tiling does not guarantee clusters to be composed of only one uniform region; though clusters composed of two or more partial regions may degrade the overall SP extraction accuracy. To overcome this imperfection due to initialization, initial grid should be refined. For refinement of the initial grid, two different approaches are proposed in the following sub-sections.

The overall algorithm is summarized in Table 2.1 as a pseudo-code. The executable of this algorithm can be obtained from http://www.kutalmisince.com/icip2015.

Table 2.1: LASP Algorithm pseudo-code

1. *Initiate the cluster centers as a honey-comb,*
2. *Assign each pixel to nearest cluster center on the image plane,*
3. *Perform refinement of the initial grid,*
4. *Initiate spectral distributions of clusters and boundary pixel list,*
5. *For the maximum number of iterations:*
   - i. *Update the labels of the pixels on the boundary list using (2.9) except the settled clusters,*
   - ii. *Update spatial and spectral distributions,*
   - iii. *Update settled clusters and boundary pixel list,*

2.2.1.1 Predefined Re-segmentation

The proposed re-segmentation process is similar to the method proposed in [19]. Given the initial clusters, for five different combinations of sub-clusters (as in Figure 2.3), contrast, (m), between the sub-clusters is computed for each cluster as follows:

$$m = \begin{cases} \left(\bar{I}_1 - \bar{I}_2\right)^2 & r = 2 \\ \left(\left(\bar{I}_1 - \bar{I}_2\right)^2 + \left(\bar{I}_1 - \bar{I}_3\right)^2 + \left(\bar{I}_2 - \bar{I}_3\right)^2\right)/3 & r = 3 \end{cases} \tag{2.10}$$

For each cluster, the combination which gives the highest contrast (2.10) is selected. Then, starting from the cluster having the highest contrast, new clusters are generated via exploiting the selected sub-cluster combination. This re-segmentation process is continued until the required number of SPs is achieved or the highest contrast is below a given threshold (T_{con}). In this method, moving the cluster center to the lowest gradient position as defined in [15] is not required. The re-segmentation approach is quite fast and improves robustness against the initial grid.

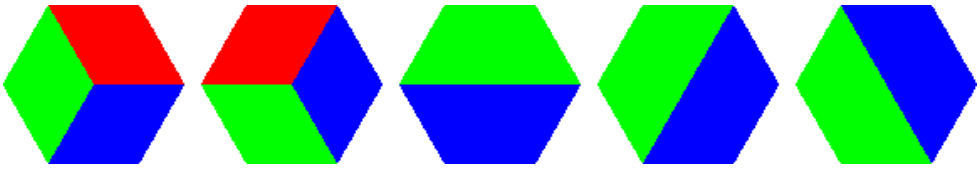

Figure 2.3: Five different combinations for predefined re-segmentation.

2.3 Experiments

In this section, performance metrics for evaluation of SP extraction performance are presented, quantitative analysis of the proposed method is provided which is followed by the comparative tests with the state-of-the-art methods in terms of computational complexity and accuracy. Proposed approach is compared to the following popular approaches, TP [15], SuTP [16], SLIC [18], SEEDS [19], Graph-based (GB) [2] and Structure Sensitive Geo (SS-Geo) [23]. The experiments are performed on the Berkeley segmentation database [24], which contains 300 images (481x321) with 24bit

RGB channels. The algorithms are compared in terms of execution time and accuracy by the use of boundary recall and bleeding error. The experiments are conducted on a PC with i5 3.2GHz CPU and 4GB RAM.

2.3.1 Performance Metrics

In order to measure the performances of the SP extraction algorithms, bleeding error and boundary recall metrics are utilized [15]. The bleeding error is the complement of the precision for the segmentation via SPs when the recall ratio is equal to one. To obtain the bleeding error ratio, for each object m, the difference between the total area of the SPs overlapping with the object and the area of the object is normalized with the area of the object. The overall bleeding error is obtained by averaging the individual rates for all objects. In a more formal way, bleeding error is defined as:

$$m = \frac{1}{M} \sum_{m=1}^{M} \frac{\sum_{i|\Omega_i \cap \Omega_m \neq =0} |\Omega_i| - |\Omega_m|}{|\Omega_i|} \tag{2.11}$$

where Ω_i is the region of support for SP i, $|\Omega_i|$ denotes the number of pixels in the region of support, and M is the total number of objects (segments) on the frame.

Boundary recall is proposed to approximate the recall rate. Boundary recall is defined as the ratio of the object boundary pixels in a small neighborhood (i.e. 2 pixels) of the SP boundaries over the total number of object boundary pixels.

2.3.2 Simulations on the Proposed Approach

In this section, the effect of channel normalization (local adaptation) and refinement of initial grid on the bleeding error and boundary recall performance is analyzed by varying spatial proximity parameter (λ). Bleeding error and boundary recall of the proposed method are shown in Figure 2.4 and Figure 2.5, with and without the channel normalization (local adaptation), for various number of SPs. For the same convexity weight, channel normalization reduces bleeding error as well as the boundary recall. However, as the number of SPs increases, boundary recall for both cases converges to the same point, whereas the improvement in bleeding error provided by local adaptation is preserved.

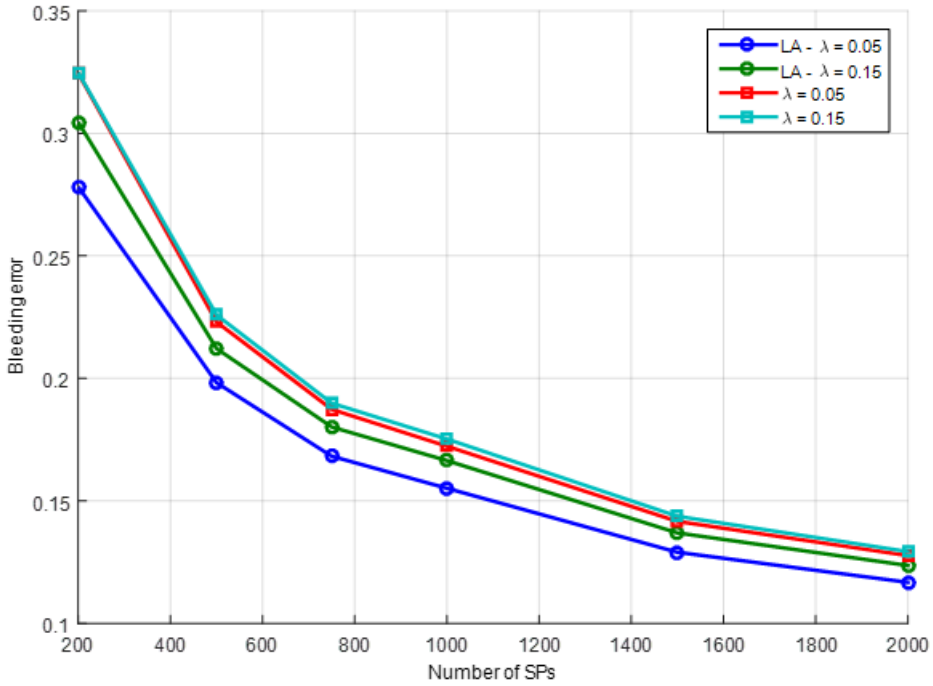

Figure 2.4: Bleeding error for local adaptive (LA) and non-adaptive SP extraction for different number of SPs and different λ parameters. λ is normalized with the global average of SP area.

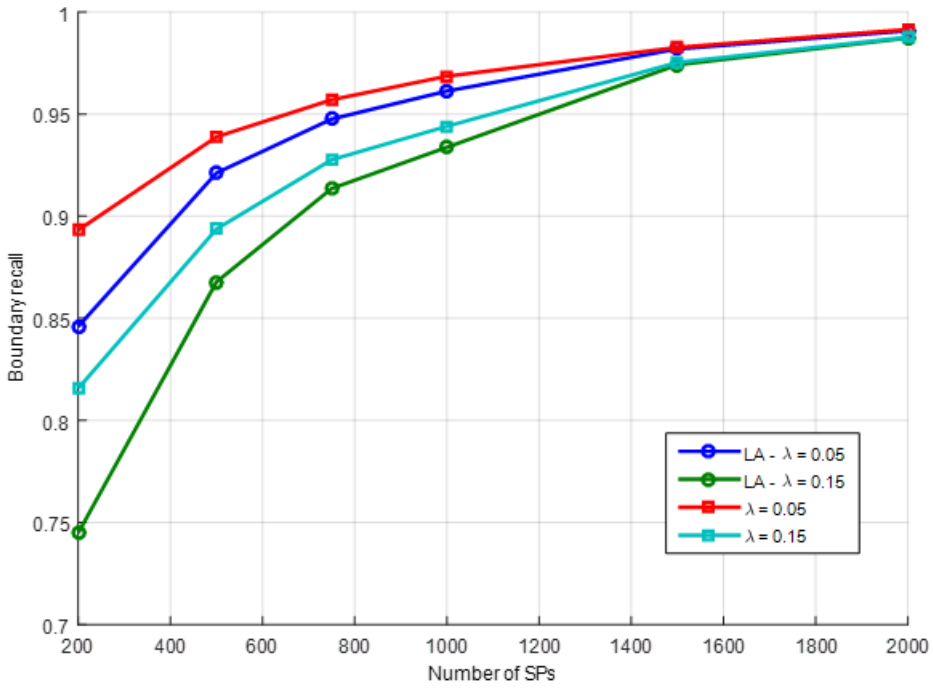

Figure 2.5: Boundary recall for local adaptive (LA) and non-adaptive SP extraction for different number of SPs and different λ parameters. λ is normalized with the global average of SP area.

2.3.3 Comparative Tests against state-of-the-art

LASP is compared against state-of-the-art SP extraction techniques in terms of bleeding error and boundary recall. Results of different methods with respect to number of SPs are presented in Figures 2.6 and 2.7. As shown in Figure 2.6 bleeding errors are quite close to each other, except SS-GEO and LASP follows SS-GEO with SEEDs. In Figure 2.7, it is shown that LASP outperforms other methods significantly, while providing a slight improvement over SEEDs.

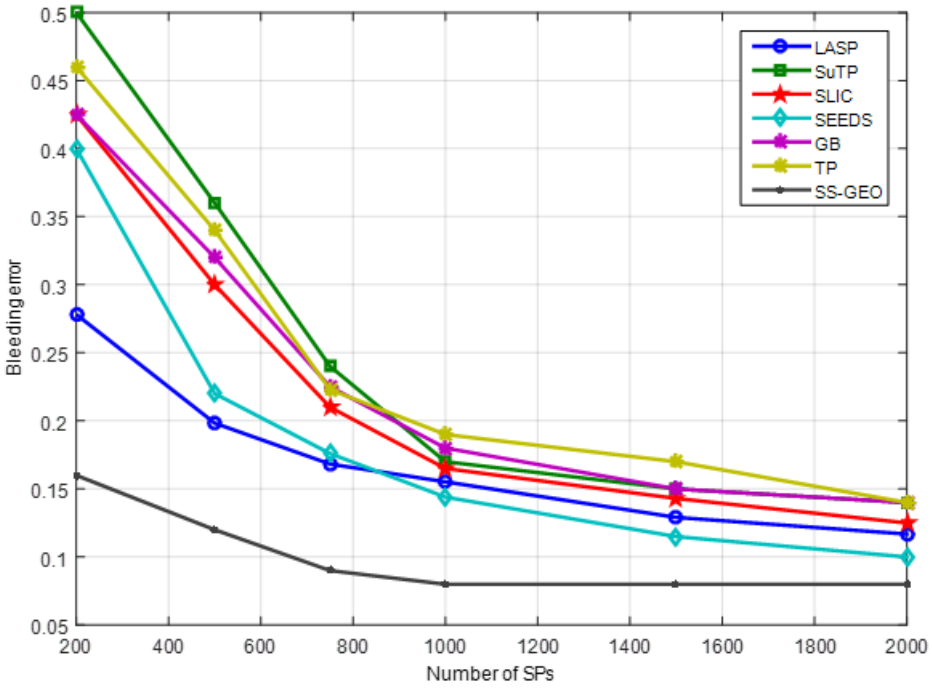

Figure 2.6: Bleeding error for different methods with respect to the number of SPs.

The comparison of the computational time is presented in Figure 2.8 for LASP, SuTP, SLIC, SEEDs and GB. Starting from more convex (hexagon) clusters, terminating update of settled clusters and visiting only boundaries pixels, LASP is hardly affected by the number of SPs and almost two times faster than SLIC, whereas timing of the SuTP and SEEDS increase with the increasing number of SPs. In order to achieve a similar performance, SEEDs needs 6 times more computational time compared to LASP. Since SS-Geo [23] and TP [15] are reported to have much higher execution time, thus they are not included in this figure.

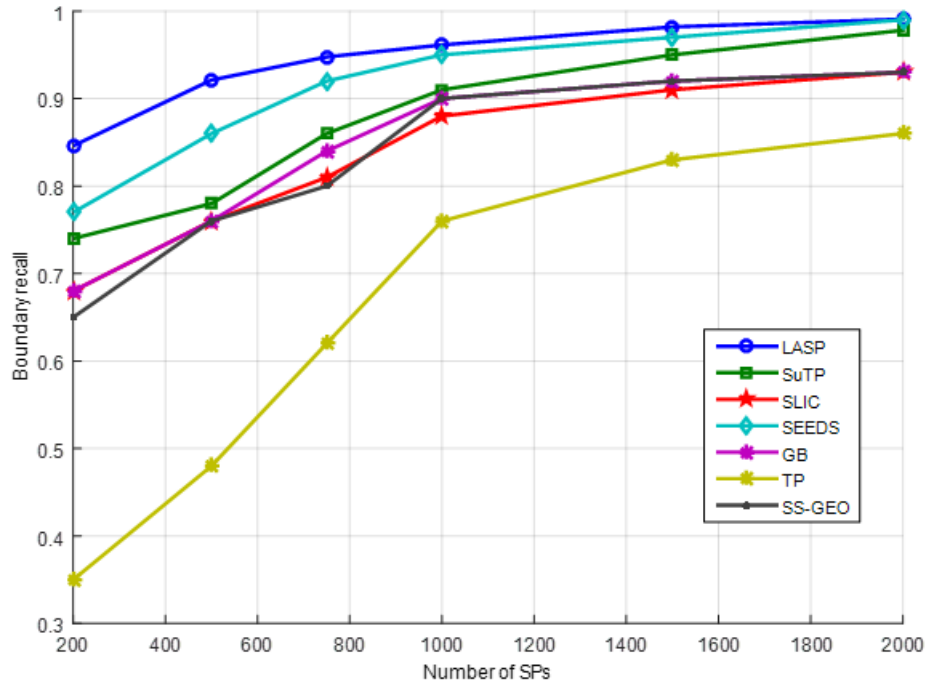

Figure 2.7: Boundary recall for different methods with respect to the number of SPs.

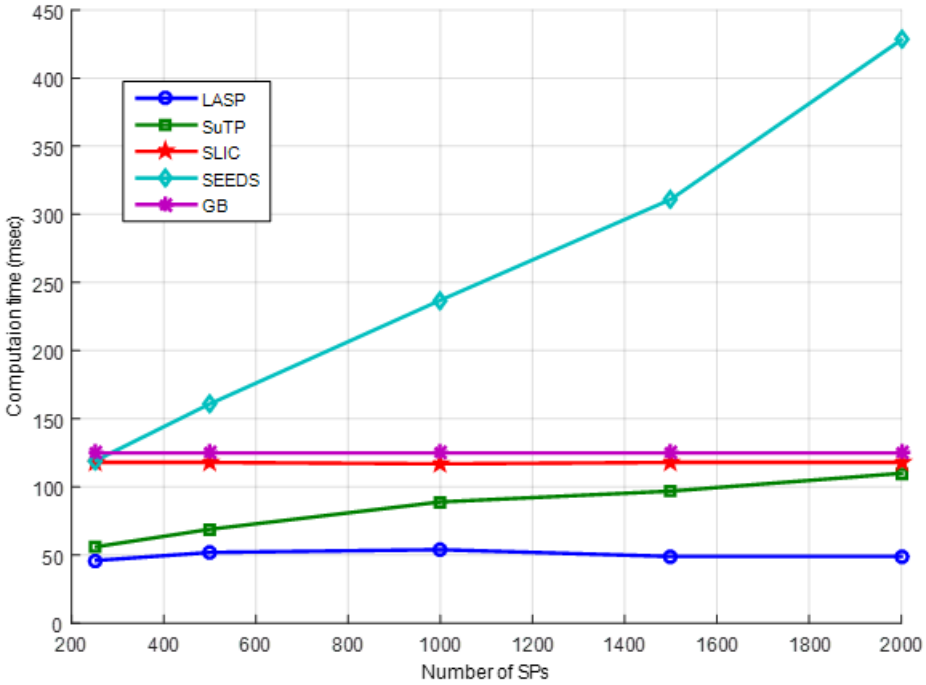

Figure 2.8: Computation time (msec) for different methods with respect to the number of SPs.

2.4 Conclusion

In this chapter, a novel and efficient method for SP extraction is presented. Parameter dependency on similar SP extraction algorithms is avoided by iteratively estimating local spectral variances. Algorithm results in regularly shaped and distributed SPs in the textured regions while preserving the accuracy on the low contrast regions. The initiation of the clusters as hexagons helped to obtain more convex SPs with reduced complexity, whereas detecting the settled SPs and stopping the update around these SPs reduced the computation time. Moreover, refinement of the initial grid by different methods has increased the overall accuracy of the method. LASP is shown to outperform the most efficient methods in terms of both accuracy and execution speed, providing a good alternative for state-of-the-art.

CHAPTER 3

MOTION ESTIMATION FOR SUPERPIXEL REPRESENTATIONS

Motion estimation is a well-studied problem in computer vision; starting from the fundamental algorithms around 80's [25, 26], various methods have been proposed during this time [27, 7, 28, 29, 30, 31]. All of these methods are mainly based on image gradients forced by brightness consistency [6]. The so-called brightness consistency constraint provides an underdetermined set of equations which results in the necessity of regularization terms forcing the motion filed to be smooth. Lack of strong gradients (i.e. aperture effect) is a major problem for these methods [6], which also results in the necessity of regularization terms. The utilization of the image gradients is problematic when the regularization term forces a smooth motion field, since the strongest gradients are mostly on the object boundaries where the discontinuities of the motion filed might occur. Hence, the segmentation information, which provides the object boundaries, is quite valuable in motion estimation problem, thus it should be employed in regularization. When the segmentation results are utilized for the regularization, quite successful results are obtained [7].

For relatively small clusters (i.e. superpixels), the translational motion assumption is generally valid, and if the consistent segmentation results were available for such clusters, then the motion of each cluster would be easily expressed by the displacement of the cluster center. If the displacements of SPs were available, this information could be utilized by any object segmentation algorithm working on superpixels (SPs). The dense motion estimation results can also be obtained by assigning the same displacement vector to each cluster to its member pixels. This motion field might be used as

an initial estimate and can be refined with well known dense motion estimation methods. In this case, the uniform/constant displacement over the cluster might be utilized as a regularization term, or the neighborhood relations on the cluster boundaries can be discarded during regularization. However, consistent segmentation results would require at least a sparse motion field. Starting from the motion field solution might be a good alternative. If the over-segmentation is performed on one of the frames, this information can be exploited to solve the underdetermined motion estimation problem.

In this chapter, an SP-based occlusion-aware layered motion estimation method is presented. The chapter is organized as follows: in the first section, general formulation of the optical flow problem is introduced and the related work on the motion estimation problem is reviewed. In the second section, the proposed method is presented which is followed by the experiments in the third section, and the conclusions on the fourth section.

3.1 Related Work

In this section, the previous related work on motion estimation problem is reviewed. Following the classical methods and their extensions, SP-based solutions and a brief discussion on these solutions are presented.

3.1.1 Classical Methods

The solution of the optical flow mostly depends on the brightness consistency equation (3.1), which requires constant illumination between frames and the observed surface to be Lambertian [6].

$$I(x + u_x, y + u_y, t + \Delta t) = I(x, y, t) \tag{3.1}$$

For a discrete time signal, the optical flow constraint equation is expressed as:

$$I_k(x + u_x, y + u_y) = I_{k-1}(x, y) \tag{3.2}$$

where k is time (frame) index. By using first order Taylor series expansion of the left-hand side, (3.2) can be linearized, resulting in an approximate equality, namely

optical flow constrained equation:

$$I_k(x,y) + u_x \frac{\partial I_k(x,y)}{\partial x} + u_y \frac{\partial I_k(x,y)}{\partial y} \cong I_{k-1}(x,y) \tag{3.3}$$

(3.3) can be expressed as a set of linear equations:

$$[I_h(x,y)\ I_v(x,y)] \begin{bmatrix} u_x \\ u_y \end{bmatrix} - I_t(x,y) = 0 \tag{3.4}$$

where $I_h(x,y)$, $I_v(x,y)$ and $I_t(x,y)$ are horizontal, vertical and temporal derivatives at point (x,y), respectively. For the horizontal and vertical derivatives central difference is used in common and the temporal derivative is defined as the difference between the current and the previous frames (3.5):

$$\begin{aligned} I_h(x,y) &= \left(I_k(x+1,y) - I_k(x-1,y)\right)/2 \\ I_v(x,y) &= \left(I_k(x,y+1) - I_k(x,y-1)\right)/2 \\ I_t(x,y) &= I_{k-1}(x,y) - I_k(x,y) \end{aligned} \tag{3.5}$$

For a c channel multi-spectral image, the partial derivatives at position (x,y) are $c \times 1$ column vectors. In one channel case (gray-level image), there are two unknowns for each pixel, subject to single constraint equation (3.4), which makes the problem under-determined.

Let $\boldsymbol{x}$ denote the position on image plane and $\boldsymbol{u}$ be the motion vector. If the motion vector is constant over a region Ω, then the problem can be solved by minimizing the following expression:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \sum_{\boldsymbol{x}\in\Omega} \left\| I_k(\boldsymbol{x}+\boldsymbol{u}) - I_{k-1}(\boldsymbol{x}) \right\|^2 \tag{3.6}$$

If there is a constraint on the motion vector, restricting the deviation from an expected value, $\bar{\boldsymbol{u}}$, with a weight, λ, then the minimization problem is expressed as:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \sum_{\boldsymbol{x}\in\Omega} \left\| I_k(\boldsymbol{x}+\boldsymbol{u}) - I_{k-1}(\boldsymbol{x}) \right\|^2 + \lambda \left\| \boldsymbol{u} - \bar{\boldsymbol{u}} \right\|^2 \tag{3.7}$$

Applying Taylor series expansion to I_k in (3.7), rearranging the equation and removing the constant terms, following minimization problem can be obtained:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \boldsymbol{u}^T \left(\lambda I + A\right) \boldsymbol{u} - 2\boldsymbol{u}^T \left(\lambda \bar{\boldsymbol{u}} + b\right) \tag{3.8}$$

where the structure tensor A and the error vector b are defined as:

$$A = \begin{bmatrix} \sum_{\boldsymbol{x}\in\Omega} \|I_h(\boldsymbol{x})\|^2 & \sum_{\boldsymbol{x}\in\Omega} I_h^T(\boldsymbol{x}) I_v(\boldsymbol{x}) \\ \sum_{\boldsymbol{x}\in\Omega} I_h^T(\boldsymbol{x}) I_v(\boldsymbol{x}) & \sum_{\boldsymbol{x}\in\Omega} \|I_v(\boldsymbol{x})\|^2 \end{bmatrix} \tag{3.9}$$

$$b = \begin{bmatrix} \sum_{\boldsymbol{x}\in\Omega} I_h^T(\boldsymbol{x}) I_t(\boldsymbol{x}) \\ \sum_{\boldsymbol{x}\in\Omega} I_v^T(\boldsymbol{x}) I_t(\boldsymbol{x}) \end{bmatrix} \tag{3.10}$$

Taking the partial derivatives with respect to $\boldsymbol{u}$ in (3.8) and equating to zero, the following least squares solution can be obtained:

$$\hat{\boldsymbol{u}} = (\lambda I + A)^{-1} (\lambda \bar{\boldsymbol{u}} + b) \tag{3.11}$$

which is the *general classical solution* for the optical flow equation. If the weight of the regularization term is zero, and the image is single channel, then the solution becomes the Lucas-Kanade (LK) solution [26]:

$$\hat{\boldsymbol{u}} = -\begin{bmatrix} \sum_{\boldsymbol{x}\in\Omega} I_h^2(\boldsymbol{x}) & \sum_{\boldsymbol{x}\in\Omega} I_h(\boldsymbol{x}) I_v(\boldsymbol{x}) \\ \sum_{\boldsymbol{x}\in\Omega} I_h(\boldsymbol{x}) I_v(\boldsymbol{x}) & \sum_{\boldsymbol{x}\in\Omega} I_v^2(\boldsymbol{x}) \end{bmatrix}^{-1} \begin{bmatrix} \sum_{\boldsymbol{x}\in\Omega} I_h(\boldsymbol{x}) I_t(\boldsymbol{x}) \\ \sum_{\boldsymbol{x}\in\Omega} I_v(\boldsymbol{x}) I_t(\boldsymbol{x}) \end{bmatrix} \tag{3.12}$$

However, in order to have a solution, the matrix (structure tensor) on the right-hand side should be well-conditioned, which is satisfied only for some specific blocks. LK (mostly denoted as KLT as well) solution can be applied to the blocks, whose structure tensor have large eigenvalues. Such blocks contain Harris corners [32] or good features to track [33].

For a single channel image, if the region of support for the motion vector, Ω, is defined as a single pixel, then the solution can be expressed as:

$$\hat{\boldsymbol{u}}(\boldsymbol{x}) = \begin{bmatrix} \lambda + I_h^2(\boldsymbol{x}) & I_h(\boldsymbol{x}) I_v(\boldsymbol{x}) \\ I_h(\boldsymbol{x}) I_v(\boldsymbol{x}) & \lambda + I_v^2(\boldsymbol{x}) \end{bmatrix}^{-1} \left(\lambda \bar{\boldsymbol{u}} - \begin{bmatrix} I_h(\boldsymbol{x}) I_t(\boldsymbol{x}) \\ I_v(\boldsymbol{x}) I_t(\boldsymbol{x}) \end{bmatrix} \right) \tag{3.13}$$

If an iterative solution is applied and $\bar{\boldsymbol{u}}$ is selected as the average of four immediate neighbors of the pixel, then the solution becomes equal to the iterative Horn-Schunck solution [25]:

$$\boldsymbol{u}^{(n+1)} = \bar{\boldsymbol{u}}^{(n)} - \frac{1}{\lambda + I_h^2 + I_v^2} \left(\begin{bmatrix} I_h^2 & I_h I_v \\ I_h I_v & I_v^2 \end{bmatrix} \bar{\boldsymbol{u}}^{(n)} + \begin{bmatrix} I_h I_t \\ I_v I_t \end{bmatrix} \right) \tag{3.14}$$

where (n) indicates the iteration number and spatial indexes are omitted for simplicity.

3.1.2 Extensions of Classical Methods and Alternative Approaches

Following the first parametric solutions for the optical flow problem, various methods have been proposed to solve the underdetermined optical flow constraint equation (3.1). Among them, one of the most important efforts is the Bayesian solution proposed by Konrad [27], expressing the motion estimation problem in a stochastic formulation. In this study, in addition to the motion field in classical formulation, a novel line field is first proposed that allows preserving the discontinuities in the motion field. The solution of this stochastic problem formulation yields to maximum a posteriori (MAP) estimate of motion field. Later with the addition of the occlusion field [34], utilization of the line and occlusion fields become common in motion estimation problem [35].

According to [6], in a probabilistic point of view, the penalty function (also known as the data term in optical flow formulation) given in (3.6) correspond to independent and identically Gaussian distributed optical flow gradient which is not the case in general. Therefore, one of the most popular penalty functions for data term is L1 norm, which yields to minimization of the total variation. L1 norm is also reported to preserve discontinuities better [29]. In some studies, rather than using the constant brightness assumption, illumination changes, blur and other changes in appearance are also considered [6] which makes the problem more underdetermined.

Using the anisotropic smoothness is reported to have a better performance [28] than the isotropic smoothness constraint defined as in [36]. Another alternative for the prior term is to use rigidity assumption and epipolar geometry [29].

According to [6], the best performing methods, utilizes L1 norm, anisotropic smoothness [28, 31] and epipolar geometry [30]. Rather than complex cost functions or priors due to the camera geometry, simple modifications, such as median filtering on classical methods [36] are also reported to enhance the performance [37].

Another remarkable solution is proposed by De Haan [38], which belongs to a completely different family of solutions. This solution is based on block matching and recursive search, resulting in a quite fast algorithm while minimizing the reconstruction errors as well.

3.1.3 Superpixel-based Solutions

The methods performing motion estimation and/or temporally consistent segmentation with SPs can be classified into three groups. The first group of methods performs joint SP extraction and motion estimation, which will be denote as *joint solutions* [8, 39]. The second group performs SP extraction on each frame independently which is followed by the association of the SPs between the frames; those methods can be called as *independent solutions* [40]. The third group extracts SPs on the first frame and then propagates those SPs to other frames by motion estimation, which will be called as *propagating solutions* [41, 42]. The second group mostly deals with the segmentation problem, while the third group is used for accurate motion estimation; whereas the first group handles the joint problem.

Temporal SPs (TSPs) are first defined in [39] and difference between TSPs and supervoxels are clearly stated in this study: A supervoxel is a volumetric cluster similar to TSP; however, TSPs are specifically designed to preserve the point-to-point matching throughout the video. In other words, supervoxel representation is shown to be efficient for 3D volumetric data, where the third dimension is spatial; however, when the third dimension is temporal (video), a different solution is needed in which the point correspondences between frames are also preserved.

Zitnick [8] purposes a joint method for segmentation and motion estimation. One of the main motivations for the joint motion estimation and segmentation is stated as *"the inability to recover from segmentation errors of pre-computed color segments"*. Even if the term SP is not used, generated segments are quite similar to SPs, except having a regular size and a convex shape. As a joint solution, the joint segmentation and motion estimation are performed iteratively in this approach. Displacement over a segment is assumed to be constant. At the first step regions are placed to their estimated positions and pixel-region memberships are updated. At the second step estimated positions of the regions are updated where displacement of a region is defined as displacement of its center. These steps are repeated iteratively until convergence. For the occluded regions and possible estimation errors, each SP is allowed to take one of its neighbor's motion vector as well as its own displacement, as in 3DRS algorithm [38].

In [39], another joint solution for motion estimation and SP extraction is proposed. In this study, it is assumed that there is an underlying Gaussian random process generating the pixels in a SP. The mean parameter of this random process is extracted from the input image; however, variance parameter is selected as a hyper parameter. The spatial correlation is discarded in this solution. Once the pixel to SP membership is given, the overall observation probability is specified since the parameters of Gaussian process are clearly defined. Therefore, for a single SP, the algorithm tries to find a solution, which minimizes the variance over a SP. For a single frame solution, the objective is to minimize the sum of negative log likelihood of all SPs. Minimization is performed by iteratively applying pixel label change, SP merge and SP split steps until no possible movement left to reduce the cost. Parameters of the appearance process are assumed to deviate from the parameters in previous frame slightly without any correlation between frames. It is assumed that there is another Gaussian random process generating the SP position. This process is used to punish the deviations from the expected SP position, which is obtained by updating previous SP position with a smooth motion field. Using these Gaussian processes, another likelihood function is defined to be maximized for consistent segmentation. In this method, the displacements of SPs are neither expressed nor solved explicitly. Since the occlusions, dis-occlusions and deformations are not modeled, it is hard to conclude about the motion of a SP. The authors [39] report that the algorithm is sensitive to the initialization of the motion field; therefore, the motion field is initiated with user interaction. Expressing the displacement of a SP as the sum of camera movement (globally smooth motion) and self-displacement of that SP (a linear model with n^{th} order zero derivative, i.e. constant velocity, acceleration etc.) would better model the dependencies of SP position between frames for this approach.

In [40], an independent solution for video segmentation by utilizing SP flows is proposed. In this method, SPs are generated for each frame independently and the SPs in adjacent frames are matched.

Utilization of the well-known *mean-shift algorithm* for tracking purposes is first proposed in [43], and mean-shift algorithm is proven to converge by [44]. In [41], it is also shown that the mean-shift approach is applicable to SP motion estimation

problem as well, and a SP-based motion estimation method for estimating large displacements is proposed. In this study, however, a clear mathematical reasoning for the proposed energy function is missing. Even if the proposed method is able to handle the large displacements, it is not considered that the large displacements would yield large occlusions, and the occlusions are unfortunately not modeled. The occluded regions are estimated through a forward-backward solution comparison; however, since the smoothness term in the energy function would yield the erroneous estimates of occluded SPs to propagate through the uniform regions, the accuracy of the solution is reduced near to the occlusion regions.

A SP-based propagating solution for motion estimation is first proposed in [42]. The proposed method starts with SPs on the previous frame obtained via SLIC and a dense motion field, then performs MAP estimation for the layer ordering and the motion field. Different from the conventional approaches, which try to solve the motion, line and occlusion fields, in this approach, when the SPs are available the line field and the occlusions are completely defined by *layer ordering*. Therefore, the authors [42] state that the overall problem is finding the most likely layer orders and motion field for a given a pair of images. Hence, their proposed algorithm starts with a high quality dense motion field; given the estimated motion field, at the first step for each pixel probability of being occluded is determined. Then, the most probable motion states are selected and candidates are sampled around these initial estimates and loopy belief propagation is utilized to determine the most likely set of motion vectors. These two steps are performed iteratively until the convergence. In their study [42], the parameter selection and computational complexity issues are not discussed. However, requiring a high quality dense solution as an initial estimate and performing belief propagation with randomly sampled particles as an inner loop step, computational complexity is expected to be relatively high. Moreover, there are five different terms composing the energy function, each having a user defined weight. Such weights are expected to have a significant effect on the performance of the algorithm. The sampling around the most possible motion vectors can be considered similar to the 3DRS algorithm [38].

3.1.4 Discussion on Superpixel Based Solutions

Independent solutions over SPs might be applicable to video object segmentation problems, but it is difficult to estimate a precise motion field with such methods. Since SP extraction is an underdetermined problem, the initial conditions determine the SP solution where the algorithm will converge. Moreover, even if there were a global optimum for SP extraction, since SP extraction methods mostly depend on gradient ascent approaches, there would be a local minimum problem. Matching the underdetermined solutions obtained with different initial estimates (or the same initial estimates on different frames) is a challenging problem. If the task is to perform video object segmentation, then matching a group of SPs on one frame to a group of SPs on another frame might be an alternative. However, instead of trying to solve this matching problem, trying to extract SPs consistently would be easier under certain assumptions, such as the brightness consistency and small displacement for SPs.

When the segmentation and motion estimation problems are considered, it is quite clear that a joint solution is necessary, since segmentation output helps to constrain the motion field and the motion field is quite useful for segmentation. However, this approach does not necessarily apply to SP extraction; since SP extraction (or in general over-segmentation) is an underdetermined problem, it is possible only consider the optimal solution for the given initial conditions, as it is obvious that there are many solutions which may achieve the same segmentation accuracy. In other words, a joint solution does not necessarily enhance the over-segmentation performance. Moreover, since SP extraction methods, such as LASP or SLIC are quite successful, there is no need to update the pixel to SP memberships simultaneously along all frames of a video.

Starting with the SPs in the first frame and updating the pixel-SP memberships throughout the video, a computationally efficient propagating solution can be achieved. Such an approach corresponds to stating an initial condition (SPs on the first frame) for TSPs. Utilization of a propagating solution makes it applicable for online video processing and easier to apply well-known Kalman filtering techniques for position estimation [45].

SPs provide a significant information for the line and occlusion fields throughout the layer order; therefore, such an approximation should be kept. However, discarding this valuable information and utilizing another motion estimation algorithm for initiating the motion field [39, 42] would result in a computationally complex algorithm and bring the problems caused by the algorithm utilized for the initial estimate. Rather than sampling the motion vectors randomly for global optimization [42], gradient descent methods, such as [41], might be utilized to obtain high quality, local minimum generating particles. However, for the gradient ascent approach, utilizing the SP level gradients [41] rather than the pixel level gradients reduces the accuracy of the motion estimates.

Assuming that the pixels in an SP are generated through a Gaussian random process, which is changing slowly between frames, is quite reasonable; so this assumption should be kept. Expressing the motion of an SP as a single translation is quite effective and shown to be valid; hence, the same assumption should be utilized in common. With this assumption, it becomes possible to make the conclusion that the displacement of an SP is equivalent to the displacement of its center unless it is occluded and the mean-shift approach [43] becomes applicable.

According to the discussion given above, an SP-based motion estimation should include the following advances and avoid the following problems:

- Solutions in [39, 42] require a high quality dense motion estimate for the solution of the problem, while an SP-based motion estimation method should not require any motion estimates.
- SP generating random processes should be utilized as in [39] and translational motion assumption for SPs should be kept [8].
- Adding the layer orders to the problem as an unknown field, as in [42], is expected to help preserving the discontinuities better and handle the occlusions, while gradient descent approaches, such as [41], suffer from occlusions and motion discontinuity boundaries, and [39] is not able to provide explicit optical flow solution due to the lack of an occlusion model.
- One-to-one matching of SPs of two different frames is not possible for the

methods that apply SP extraction independently in two frames, such as [40] which works with multiple hypothesis to solve this ambiguity, while an SP-based motion estimation method should provide one-to-one matching for SPs.

- For a computationally efficient solution, the gradient descent methods should be utilized for updating the motion field for each motion vector candidate and the global optimization should be performed with some particles, while [42] samples the motion vectors around the most possible states and performs belief propagation with these particles iteratively.
- In order to have high accuracy motion estimates, the gradient descent update should utilize the pixel level gradients rather than the SP level gradients in [41].

3.2 Proposed Method

The two frame motion estimation problem is defined as finding the optical flow field between two different samples taken from a scene. In this section, the two frame motion estimation problem on temporal samples (the previous and the current frames) is considered; however, the whole formulation is also applicable to the spatial samples (i.e. the left and the right frames for stereo imaging).

Occlusions and motion discontinuity boundaries are two of the main challenges of the motion estimation problem. Pixel level discontinuity and occlusion handling is proposed in various studies [7, 28, 29, 30, 31]; however, pixel level labeling of occlusion and line fields result in a huge search space, even if the most of the occlusion and motion discontinuity combinations are invalid. In order to avoid being trapped in local minimum in this huge search space, more complex algorithms are required.

A successful SP extraction algorithm results in small regions, ideally and most of the time, each including the pixels from a single semantic object, which makes it possible to perfectly define the object region as the union of SPs. Since the occlusions and motion discontinuities mostly occur at object boundaries, SP boundaries restrict their locations, so occlusions can be defined at SP level. SPs are small regions; therefore, it is easier to define a valid parametric motion model for an SP. In the proposed method, it is assumed that SPs are rigid and the motion of each SP can be represented with a

single translational displacement.

A layer order can be defined as the relative depth of the neighboring SPs. When the layer order is available, then the motion discontinuity boundaries should also be available; and when the motion field and layer order are given, the occluded regions are completely defined. Therefore, for SP level motion estimation problem, the layer order can easily replace the occlusion and line fields defined for the pixel level motion estimation problem.

For temporally consistent SP extraction, given the previous frame and the SPs on the previous frame, the task is to estimate the pixel to SP membership function in the current frame such that the point-to-point matching between the frames is preserved. Defining the motion estimation problem as finding the occlusion regions and the SP positions on current frame for the SPs given in the previous frame, and assuming that the SPs are rigid with a translational motion, the motion estimation problem becomes equivalent to find the pixel to SP membership function in the current frame which is the purpose of consistent SP extraction.

In this section, the mathematical formulation of the problem is given first. Two different algorithms are proposed for the solution of this formulation. These solutions are followed by hierarchical solution extensions of these algorithms.

3.2.1 Problem Definition

Given an image pair and SPs on one of the images, the problem is defined as finding the most probable layer order and the motion field, for the given observations [42]:

$$\left\{\hat{\boldsymbol{U}}, \hat{\boldsymbol{R}}\right\} = \arg\max_{\boldsymbol{U},\boldsymbol{R}} p\left(\boldsymbol{U}, \boldsymbol{R} | I_k, I_{k+1}\right) \tag{3.15}$$

where $\boldsymbol{U}$ is the motion field and $\boldsymbol{R}$ is the layer order. In [42], $\boldsymbol{U}$ is the dense motion field and layer order between SP i and SP j is defined as:

$$R_{ij} = \begin{cases} 1 & if\ i\ occludes\ j \\ 0 & no\ occlusion \\ -1 & if\ j\ occludes\ i \end{cases} \tag{3.16}$$

while (3.15) is factorized as [42]:

$$p\left(\boldsymbol{U},\boldsymbol{R}|I_k,I_{k+1}\right) \propto p\left(I_{k+1}|\boldsymbol{U},\boldsymbol{R},I_k\right) p\left(\boldsymbol{R}|\boldsymbol{U},I_k\right) p\left(\boldsymbol{U}|I_k\right) \tag{3.17}$$

In this factorization, the layer orders are constrained by the motion field and the motion field is constrained by the previous frame, I_{k-1}. The constraint introduced by the motion field to the layer order is quite poor; if the motion field forces the layers to overlap, than the only conclusion is $R_{ij} \neq 0$. The last term in (3.17) is the motion field constrained by the previous frame. For this probability, SP similarity is utilized in [42]; however, the similarity of SPs are actually more related to the layer order rather than the motion field. As a result of this factorization, the second term in (3.17) is marginalized over an occlusion field to find the most possible states. Once the most possible states are obtained, new particles are generated around these states (motion vectors) and the particle belief propagation algorithm is utilized for the global optimization.

On the other hand, the conditional pdf given in (3.15) can also be factorized as :

$$p\left(\boldsymbol{U},\boldsymbol{R}|I_k,I_{k+1}\right) \propto p\left(I_{k+1}|\boldsymbol{U},\boldsymbol{R},I_k\right) p\left(\boldsymbol{U}|\boldsymbol{R},I_k\right) p\left(\boldsymbol{R}|I_k\right) \tag{3.18}$$

In the proposed method, this above factorization is preferred since the previous frame gives an information about the layer ordering, rather than the motion field. The constraint introduced to the motion field by layer order is as follows: If layers are occluding each other, then the motion field will be discontinuous. Otherwise, either the changes in motion field should be smooth or they should be moving away from each other so there will be no overlap. In order to cover the moving away neighbors case, $\boldsymbol{R}$ is re-defined as in (3.19) and shown in Figure 3.1.

$$R_{ij} = \begin{cases} 1 & if\ i\ occludes\ j \\ 0 & no\ occlusion,\ same\ layer \\ \zeta & no\ occlusion,\ different\ layers \\ -1 & if\ j\ occludes\ i \end{cases} \tag{3.19}$$

Note that the values $\{-1, 0, 1, \zeta\}$ defined for R_{ij} are symbolic labels; this set is preferred to have a notational consistency with [42].

As a result of the factorization in (3.18), the layer orders are constrained by the previous image. The previous image gives an important information about the layer orders,

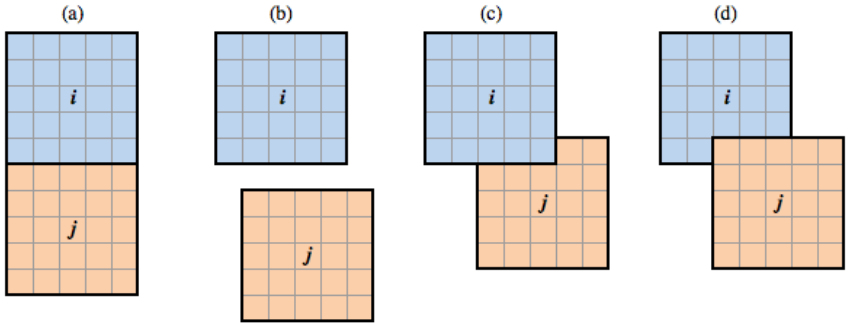

Figure 3.1: Layer order between SPs i and j. **(a)** SPs moving together, $R_{ij} = 0$ **(b)** SPs moving away from each other, $R_{ij} = \zeta$ **(c)** i occluding j, $R_{ij} = 1$ **(d)** j occluding i, $R_{ij} = -1$.

such that the neighbor regions with similar colors should belong to the same layer; they should not overlap or move away from each other. Different from [42] in which the motion field is dense, the proposed motion field is defined as a sparse motion field, obtained by the union of SP displacements. Similarly, different from [42], marginalization over an occlusion field is not performed. The terms on the right-hand side of (3.18), which are explained next, are denoted as the data term, motion prior and layer prior form left to right, respectively.

The aforementioned problem is demonstrated on Figure 3.2 with *Venus* sequence from Middlebury database [6]. In this figure, the previous frame, SPs on the previous frame (left-top) and the current frame (left-bottom) are given. Algorithm finds the most likely layer orders and motion vectors for the SPs given in the previous frame. Motion vectors of SPs from the previous frame to the current frame is shown on the previous image by red arrows. Given the ground-truth motion vectors, the layer orders are selected as shown at the right-top. Centers of the SPs belonging to the same layer are connected with cyan lines and the occluded regions on the previous frame is highlighted with red (right-top). Using the resulting layer orders and the given motion vectors, SPs of the previous frame are placed on the current frame and initial estimate for SPs on current frame obtained (right-bottom). Uncovered pixels on the current image are highlighted with green at the right-bottom.

Data Term

Motion field ($\boldsymbol{U}$) is the set of motion vectors, $\{u_i\}_{i=1}^{N}$, composed of the SP shifts from

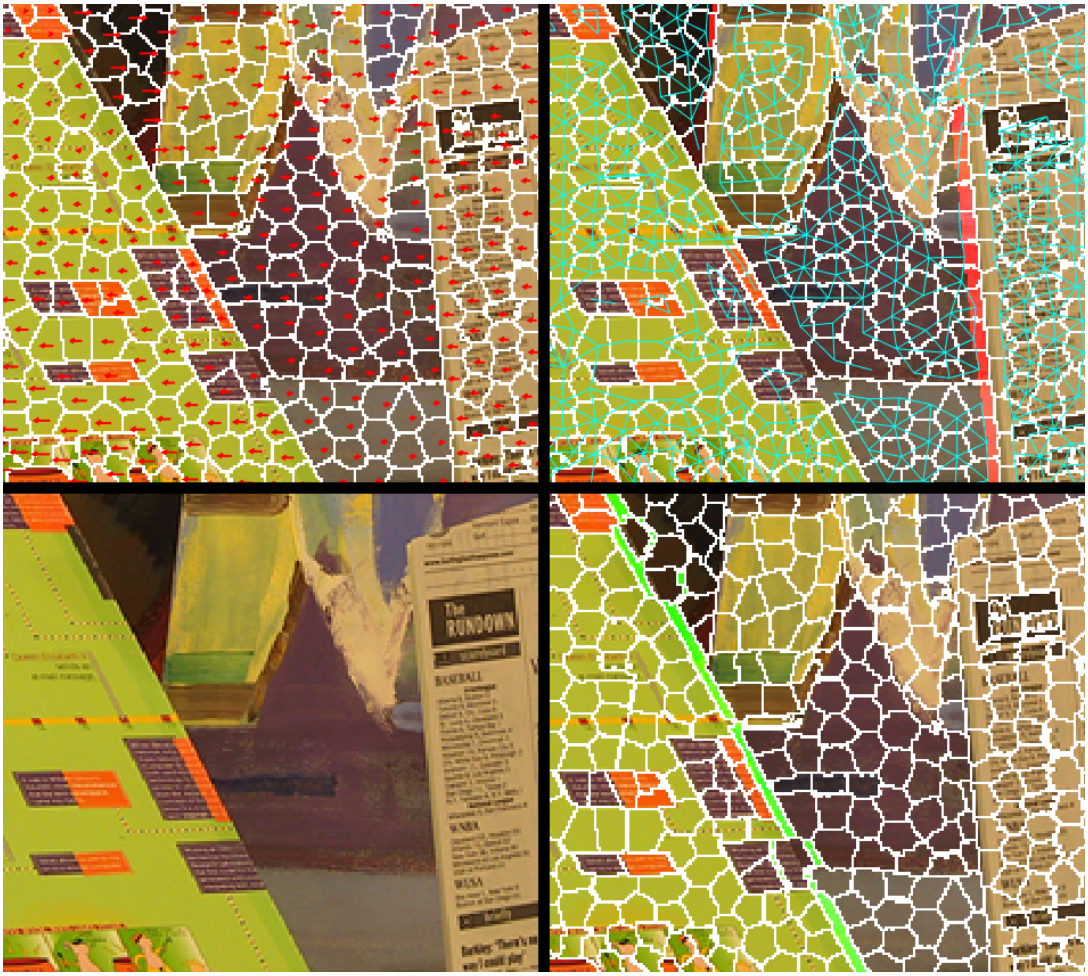

Figure 3.2: Previous frame and SPs on previous frame with ground-truth motion vectors (left-top), Occluded regions and same layer connections (right-top), Current frame (left-bottom) and SPs on current frame and uncovered regions (right-bottom).

the previous frame to the current frame to satisfy (3.20):

$$I_k(x + u_i) = I_{k-1}(x) \; \forall x \in \Omega_{i,k-1} \tag{3.20}$$

where $\Omega_{i,k-1}$ is the set of pixels assigned to SP i on I_{k-1}, or namely the region of support for SP i.

Given the motion field, SPs on the previous frame should be placed on the current frame to obtain the measurement probability. Due to the sub-pixel shifts, the pixel membership function of SP i on the current frame are not binary, while SPs should form a disjoint set by definition. To avoid the complex operations due to the sub-pixel shifts, the pixel membership function on the current frame is approximated with

integer casted motion vectors. If the occlusions are discarded, using the integer casted motion vectors, the set of pixels occupied on frame I_k by SP i is given by (3.21).

$$\Omega_{i,k} = \{x|(x - \tilde{u}_i) \in \Omega_{i,k-1}\} \tag{3.21}$$

where $\tilde{u}_i$ is the integer casted motion vector of SP i. Given the motion field ($\boldsymbol{U}$) and the layer orders ($\boldsymbol{R}$), the region of support estimate of SP i on the current frame, $\hat{\Omega}_{i,k}$, can be obtained by excluding the occluded pixels (3.22) as shown in Figure-3.3.b.

$$\hat{\Omega}_{i,k} = \{x|(x - \tilde{u}_i) \in \Omega_{i,k-1} \; and \;\; \nexists \, j \; st. \; R_{ji} = 1 \; and \; (x - \tilde{u}_j) \in \Omega_{j,k-1}\} \tag{3.22}$$

Since the occlusions are excluded, the masks obtained with (3.22) form a disjoint set.

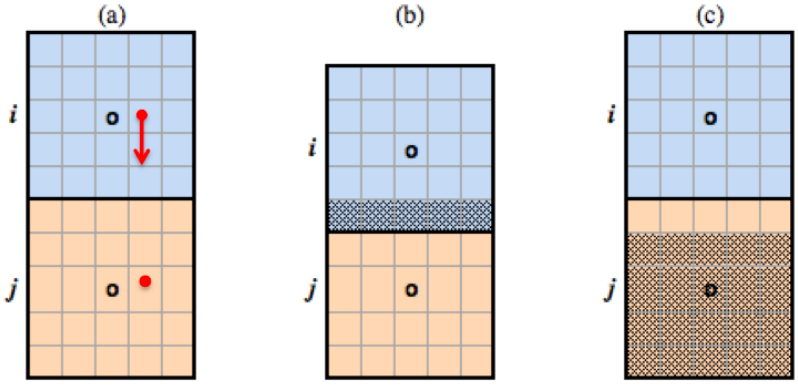

Figure 3.3: The occluded and visible regions for SP j: (a) SPs on the previous frame where motion vectors are shown in red (b) the occluded region of SP j in the current frame marked with a dot pattern (c) the visible region of SP j in the current frame is marked in the previous frame domain.

On the other hand, there will be uncovered pixels as well:

$$O = \{x| \; (x - \tilde{u}_i) \notin \Omega_{i,k-1} \forall i\} \tag{3.23}$$

The union of SP region of supports on the current frame (3.22) and the set of uncovered pixels (3.23) cover the current image completely. If the measurement noise is independent and identically distributed, the observation probability can be expressed as the product of SP observation probabilities and the observation probability of uncovered set:

$$\begin{aligned} p\left(I_k|\boldsymbol{U}, \boldsymbol{R}, I_{k-1}\right) \propto \prod_i \prod_{x \in \hat{\Omega}_{i,k}} exp\left(-\lambda \rho_d \left(I_k(x), I_{k-1}(x - u_i)\right)\right) \\ \times \prod_{x \in O} p\left(I_k(x)\right) \end{aligned} \tag{3.24}$$

where λ is a parameter of measurement noise, $p(I_k(x))$ is the probability to observe pixel x uncovered with image intensity $I_k(x)$, and $\rho_d(.)$ is the data penalty function. The probability of observing an uncovered pixel is not related to the observation in the previous frame and the location, except the image boundaries. Discarding the image boundary case, the probability of observing an uncovered pixel is assumed to be independent from the location and the observation in the previous frame and uniformly distributed for the possible values of $I_k(x)$:

$$p(I_k(x)) = 1/K \; \forall \; x \tag{3.25}$$

where K is the number of possible values for $I(x)$. Since the number of pixels in the current and previous frame are the same and SPs are assumed to be rigid, number of uncovered pixels in the current frame is equal to the number of occluded pixels in the previous frame, as demonstrated on Figure 3.2. On the bottom of the right column the uncovered pixels in current frame are highlighted with green; on its top, the occluded pixels in previous frame are highlighted with red. For a sample occlusion case, the occluded pixels in the current frame and the set of visible (or unoccluded) pixels for SP j in the previous domain are demonstrated in Figure-3.3.b and Figure-3.3.c, respectively. Visible region of support in the previous frame domain for SP i, $\Omega_{i,k-1}^{(v)}$, can be expressed as:

$$\Omega_{i,k-1}^{(v)} = \{x | x \in \Omega_{i,k-1} \; and \;\; \nexists \; j \; st. R_{ji} = 1 \; and \; (x + \tilde{u}_i - \tilde{u}_j) \in \Omega_{j,k-1}\} \tag{3.26}$$

which simply states that SP i will keep the overlapping pixels in its mask, only if it is not occluded. The occluded pixels of SP i, $\Omega_i^{(o)}$, can be obtained by excluding the visible pixels from the region of support on the previous frame:

$$\Omega_{i,k-1}^{(o)} = \Omega_{i,k-1} / \Omega_{i,k-1}^{(v)} \tag{3.27}$$

Using the set of visible and occluded pixels, the observation probability can be expressed in the previous frame:

$$\begin{aligned} p\left(I_k | \boldsymbol{U}, \boldsymbol{R}, I_{k-1}\right) \propto \prod_i \prod_{x \in \Omega_{i,k-1}^{(v)}} & exp\left(-\rho_d\left(I_k(x + u_i) - I_{k-1}(x)\right)\right) \\ & \times \prod_{x \in \Omega_{i,k-1}^{(o)}} exp\left(-\lambda_o\right) \end{aligned} \tag{3.28}$$

where $\lambda_o = \log(K)/\lambda$. Taking negative logarithm of this expression and removing the constant terms, the following objective function can be obtained:

$$J_d = \sum_i \sum_{x \in \Omega_{i,k-1}^{(v)}} \rho_d \left(I_k(x + u_i) - I_{k-1}(x) \right) + \lambda_o |\Omega_{i,k-1}^{(o)}| \tag{3.29}$$

where $|\Omega_{i,k-1}^{(o)}|$ is the number of occluded pixels for SP i. If the measurement noise is Gaussian, then the data penalty function is equal to the Mahalanobis distance:

$$\rho_d(z) = z^T \Sigma_m^{-1} z \tag{3.30}$$

where z is a $c \times 1$ column vector and Σ_m is $c \times c$ measurement noise covariance for a c channel image. The measurement noise might be assumed to be independent, identically Gaussian distributed; however, for the SPs having large spectral variances small errors in estimated motion vectors would yield to a large cost. Therefore, rather than an identically distributed Gaussian noise, for each SP image channels are assumed to be independent and variance of each image channel is utilized as the measurement noise in the corresponding channel. Under these assumptions, the final expression for data objective function is:

$$\begin{aligned} J_d(I_k, \boldsymbol{U}, \boldsymbol{R}, I_{k-1}) = \sum_i \sum_{x \in \Omega_{i,k-1}^{(v)}} \sum_c \frac{\left(I_{c,k}(x + u_i) - I_{c,k-1}(x) \right)^2}{\sigma_{c,i}^2} \\ + \sum_i \lambda_o |\Omega_{i,k-1}^{(o)}| \end{aligned} \tag{3.31}$$

Layer Prior

SPs with different colors can be assigned to the same layer, or to the different ones without any restriction; and the previous frame does not provide a significant information about the occluding SP. However, the previous frame might restrict assigning the SPs with similar colors to the different layers. For this purpose, the color distribution of SPs can be utilized to define the SP similarity, s_{ij}. The similarity of SPs i and j is defined as:

$$s_{ij} = exp\left(-(\mu_i - \mu_j)^T (\Sigma_i + \Sigma_j)^{-1} (\mu_i - \mu_j) \right) \tag{3.32}$$

where (μ_i, Σ_i) is the spectral mean and covariance of SP i. The term in the exponent measures the dissimilarity of two SPs and it is identical to Fischer's criteria [22] for a single channel image. A similar SP similarity metric is also proposed in [42], but

rather than covariance matrices of SPs, a constant term used as the standard deviation. Assigning two neighbor SPs to the different layers allows SPs to move independently, which may result in occluded or unoccupied regions. Therefore, the proposed penalty function should also be proportional to the common boundary length, which gives an information about the number of pixels to be occluded or remain uncovered, when SPs are moving independently. A good alternative for the layer order penalty is the common boundary length weighted SP similarity which is applied for the different layer assignment:

$$\rho_r^{(ij)}(r) = b_{ij} s_{ij} \delta[r \neq 0] \tag{3.33}$$

where b_{ij} is the common boundary length of SPs i and j. Once the cost of different layer assignment is defined; given the previous frame and the SPs in the previous frame, the layer prior can be obtained by applying the penalty to the neighbor SPs:

$$p\left(R|I_{k-1}\right) \propto \prod_i \prod_{j \in \Gamma(i)} exp\left(-\lambda_r \rho_r^{(ij)}(R_{ij}))\right) \tag{3.34}$$

where λ_r is the weight of the layer order penalty, and $\Gamma(i)$ is the set of neighbor SPs for SP i. Taking negative logarithm of $p(R|I_{k-1})$, following objective function can be obtained for minimization:

$$J_r(\boldsymbol{R}, I_{k-1}) = \lambda_r \sum_i \sum_{j \in \Gamma(i)} \rho_r^{(ij)}(R_{ij}) \tag{3.35}$$

Motion Prior

Neither the layer order, nor the previous frame do not clearly specify the distribution of the motion field; however, they strictly constrain the relation between the motion vectors of neighboring SPs. Given the layer order, motion field becomes independent from the previous frame, since R_{ij} clearly indicates whether the SPs are moving together or not. Obviously, if SPs are moving together ($R_{ij} = 0$) the motion field should be smooth and SPs should not overlap. If SPs are moving away from each other ($R_{ij} = \zeta$) then they certainly should not overlap. If the SPs are occluding ($|R_{ij}| = 1$), then they should overlap. Therefore, the motion prior penalty can be defined as follows:

$$\rho_u^{(ij)}(r) = \begin{cases} \lambda_s b_{ij} ||u_i - u_j||^2 & |\Omega_{ij}| = 0 \; and \; r \; = 0 \\ \infty & |\Omega_{ij}| \neq 0 \; and \; r \; = 0 \\ \lambda_{ovr} s_{ij} |\Omega_{ij}| & |\Omega_{ij}| \neq 0 \; and \; |r| = 1 \\ \infty & |\Omega_{ij}| = 0 \; and \; |r| \neq 1 \\ 0 & |\Omega_{ij}| = 0 \; and \; r \; = \zeta \\ \infty & |\Omega_{ij}| \neq 0 \; and \; r \; = \zeta \end{cases} \tag{3.36}$$

where $|\Omega_{ij}|$ is the number of overlapping pixels of SPs i and j on the current frame, and the overlapping set, Ω_{ij}, is defined as:

$$\Omega_{ij} = \Omega_{i,k} \cap \Omega_{j,k} \tag{3.37}$$

In order to force the SPs to occlude with the SPs with different colors, rather than the ones with similar colors, the number of overlapping pixels is punished with λ_{ovr} and the penalty is weighted by SP similarity. Consider the case demonstrated in Figure 3.4, where SP i occludes SP j and occluded region is demonstrated by a dot pattern. As shown in Figure 3.4.a, j is stationary and i moves towards j (motion vectors shown in red) resulting in the current frame shown in Figure 3.4.b. If the similarity of overlapping SPs is not included in the cost function, the true solution shown on Figure 3.4.c and the erroneous solution (obtained with the motion vector shown in blue) 3.4.d have an identical cost. However, when λ_{ovr} is included in the motion prior, the cost of i occluding j remains almost the same, since the similarity of i and j is close to zero, while cost of q occludes j is increased, and the smoothness between SPs having similar appearances is forced.

Once the motion penalty function is defined, the probability of motion field given the layer orders can be expressed as:

$$p\left(\boldsymbol{U}|\boldsymbol{R}, I_{k-1}\right) \propto \prod_i \prod_{j \in \Gamma_i} exp\left(-\rho_u^{(ij)}(R_{ij})\right) \tag{3.38}$$

Taking the negative logarithm of $p(U|R, I_{k-1})$, the following objective function can be obtained:

$$J_u(\boldsymbol{U}, \boldsymbol{R}) = \sum_i \sum_{j \in \Gamma_i} \rho_u^{(ij)}(R_{ij}) \tag{3.39}$$

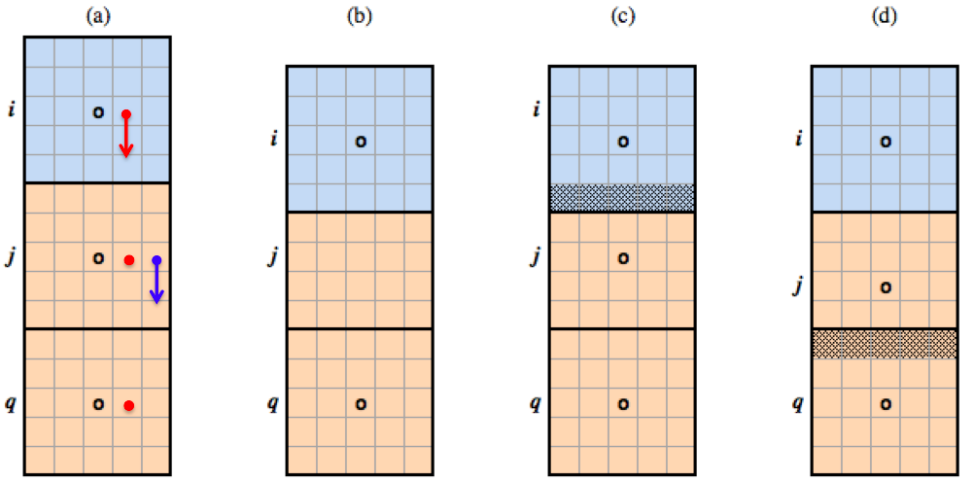

Figure 3.4: Layer order options for occluded SP j: (a) SPs in the previous frame on which the true motion vectors are shown in red and erroneous motion vector for j in blue (b) the current frame (c) j moves with q and is occluded by i (d) j moves with i and is occluded by q.

Energy Function

Starting from the maximization of conditional distribution (3.15), applying factorization (3.18) and taking negative logarithm of this expression (3.31, 3.35, 3.39); following energy function can be obtained for minimization:

$$J(I_k, \boldsymbol{U}, \boldsymbol{R}, I_{k-1}) = J_d(I_k, \boldsymbol{U}, \boldsymbol{R}, I_{k-1}) + J_u(\boldsymbol{U}, \boldsymbol{R}) + J_r(\boldsymbol{R}, I_{k-1}) \tag{3.40}$$

The MAP estimate of the motion field and layer orders is the argument minimizing the objective function:

$$\left\{\hat{\boldsymbol{U}}, \hat{\boldsymbol{R}}\right\} = \arg\min_{\boldsymbol{U},\boldsymbol{R}} J(I_k, \boldsymbol{U}, \boldsymbol{R}, I_{k-1}) \tag{3.41}$$

3.2.2 Minimization of Energy Function

The minimization of (3.41) certainly requires a global optimization procedure. The proposed objective function is differentiable with respect to the unknown motion vectors for the fixed layer orders, except the motion prior for $|R_{ij}| = 1$, but it is not differentiable for the unknown layer orders. If it were differentiable with respect to all unknowns, then the minimization could be performed with iterative Gauss-Newton

optimization, provided that the initial estimates are relatively close to the global minimum. Given the SP masks, it is possible to converge to the local minimum in a large neighborhood by using Lucas-Kanade (LK) [26] and especially with Mean-Shift (MS) [43] solutions, and for the given motion vectors MAP estimates for the layer orders can be easily obtained. In other words, starting with an initial estimate of the motion field, optimization can be performed via iterated conditional modes (ICM) method [46] which is utilized to find an approximate solution of MAP estimation problem in locally dependent Markov random fields.

Since ICM algorithm might trap to local minimum, common global optimization methods, such as belief propagation [47] or graph-cut [48], might be the alternatives. Both methods can be utilized to obtain the MAP estimate on discrete state MRFs; therefore, they can substitute each other for the MAP estimation problem [49]. Belief propagation is applicable for estimating marginal distributions also; therefore it can be employed for the minimum mean squared estimation problem. In order to apply these methods, motion vectors should be selected from a finite set. When the region of support for each motion vector is a single pixel, it is quite easy to evaluate the data term for the possible motion vectors around an initial estimate. However, this is not the case for the energy function given in (3.41).

In order to apply the belief propagation or graph-cut, consider the case in which the precision of the motion vectors are reduced (i.e. pixel accuracy) and all layer order alternatives are evaluated. Let SP i has M neighbors, then there are 4^M layer order alternatives, and if there are D alternative motion vectors for this SP, $D \times 4^M$ hypotheses should be evaluated for each neighboring motion vector alternative. In this case, there are $M \times D$ alternative neighbor motion vectors; therefore, for a frame including N SPs, $N \times M \times D^2 \times 4^M$ hypotheses should be evaluated. For a standard definition image, usually $N = 1000$, $M = 6$ and D is on the order of hundreds. It is obvious that the problem is still quite complex to solve even with reduced precision. However, when SPs are utilized for the region of support of the motion vectors, there is no need to evaluate all possible motion vectors around the given estimate, as Gauss-Newton solution can converge to the local minimum in a large neighborhood of the initial estimate.

Moreover, there is no need to evaluate all the layer order alternatives for every motion vector. Given a motion vector pair for two neighbor SPs, the layer order can be selected easily by minimizing the sum of layer and motion priors. Therefore, if the particles are generated appropriately such that the true solution is in their neighborhood, particle belief propagation may become a feasible alternative for optimization. It should be noted that the graph-cut method is also applicable to this problem, with the same MRF definition and particles.

3.2.2.1 Proposed ICM-based Solution

The proposed ICM-based solution is composed of two steps: the solution of layer orders for the given motion vectors, and the solution of motion field for the given layer orders.

ICM Solution for Layer Orders

For the given set of SP motion vectors, layer orders should be determined by maximizing (3.41). Even if the motion field is provided, the search space for this problem is still large. Moreover, the layer order selection should be *consistent*; in other words, if SP i is nearer than SP j but further than SP q, then SP q should be nearer than SP j, as demonstrated in Figure 3.5. However, if the layer order relation between each neighbor pair is solved independently, then a feasible solution can be obtained.

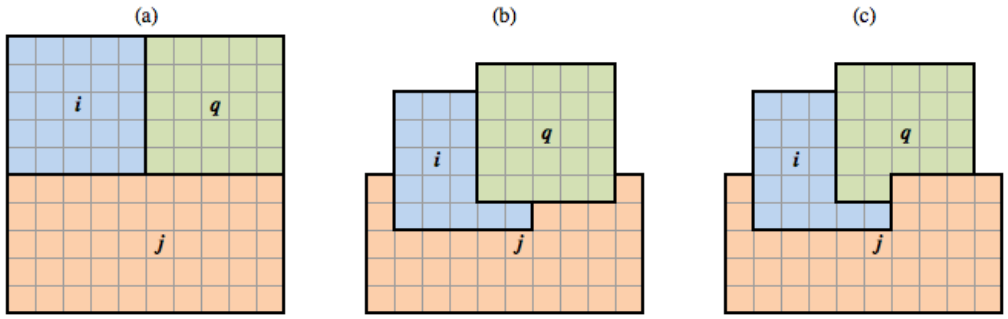

Figure 3.5: Layer order consistency: (a) SPs on previous frame (b) Consistent layer ordering (c) Inconsistent layer ordering.

If the neighbor SPs are overlapping, at the first step the occluding SP must be selected. Data penalty for SP i in the overlapping region with SP j is given by (3.42):

$$J_{d,ovr}^{(i)}(j) = \sum_{x+\tilde{u}_i \in \Omega_{ij}} \rho_d(I_k(x+u_i) - I_{k-1}(x)) \tag{3.42}$$

where $\tilde{u}_i$ is the integer casted motion vector of SP i, and Ω_{ij} is the set of overlapping pixels with SP j on the current frame as defined in (3.37). Since the prior penalty is the same for R_{ij} = 1 and R_{ij} = -1, the occluding SP should be selected according to the data cost. As the pixel-wise layer ordering is not allowed, all the pixels in the overlapping region, Ω_{ij}, should be assigned to the SP having the smaller data penalty. This results in a layer order decision which depends on prior information only, except the selection of the occluding SP for $|R_{ij}| = 1$. Since the layer order is only a function of the layer order penalty (3.33) and the motion prior penalty (3.36), layer order should be selected by minimizing the sum of these two terms:

$$\hat{R}_{ij} = \arg\min_{r} \left(\rho_u^{(ij)}(r) + \rho_r^{(ij)}(r) \right) \tag{3.43}$$

Equation (3.43) can be explicitly written as below:

$$\hat{R}_{ij} = \begin{cases} -1 & if\ J_{d,ovr}^{(i)}(j) - J_{d,ovr}^{(j)}(i) > 0 \\ 1 & if\ J_{d,ovr}^{(j)}(i) - J_{d,ovr}^{(i)}(j) > 0 \\ \zeta & if\ \rho_r^{(ij)}(\zeta) < \rho_u^{(ij)}(0) \\ 0 & o.w. \end{cases} \tag{3.44}$$

Note that the omitted terms $\rho_r^{(ij)}(0)$ and $\rho_u^{(ij)}(\zeta)$ are equal to zero.

Selecting the layer orders by (3.44) minimizes the proposed objective function for the given motion field. However, given motion vectors might be erroneous, resulting in overlapping regions for SPs on the same layer. If two SPs from the same layer are overlapping due to the errors in motion vectors, it might be impossible to reach to the true solution by using the masks updated with the decision made according to these motion vectors. Therefore, this approach might stuck to local minimum.

A more robust approach might be achieved by evaluating the data cost in the overlapping region. If SPs are actually occluding each other, then the data costs for i and j in the overlapping region would be different, indicating that the overlapping region can be reconstructed significantly better with one of the SPs. In other words, if the difference of the data costs in the overlapping region is large, then one should be occluding

the other. If the difference is close to zero, it is more likely to have an overlapping due to the errors in the estimated motion vectors. Based on this discussion, (3.44) is extended to:

$$\check{R}_{ij} = \begin{cases} -1 & if\ J^{(i)}_{d,ovr}(j) - J^{(j)}_{d,ovr}(i) > \epsilon\,|\Omega_{ij}| \\ 1 & if\ J^{(j)}_{d,ovr}(i) - J^{(i)}_{d,ovr}(j) > \epsilon\,|\Omega_{ij}| \\ \zeta & if\ \rho^{(ij)}_{r}(\zeta) < \rho^{(ij)}_{u}(0)\ and\ |\Omega_{ij}| = 0 \\ 0 & o.w. \end{cases} \tag{3.45}$$

where ϵ is the minimum required per-pixel cost difference for an occlusion decision. Once the layer order estimates are obtained, the visible region of support for each SP, $\Omega^{(v)}_{i,k-1}$, can be obtained with (3.26).

Since the given motion vectors might be erroneous, rather than a binary decision for applying the regularization, the expected value of the motion prior penalty might be minimized for the non-overlapping case. For the overlapping case, the motion prior penalty is proportional to the number of overlapping pixels; however, a closed form expression in terms of the motion vectors for this term is not available. As discussed above, difference between the data costs in the overlapping region gives an information about reliability of the neighboring motion vectors. When SPs are overlapping, but the data cost difference is small, previously defined common boundary length weighted smoothness term, $\lambda_s b_{ij}$, should be extended as below to correct the motion estimates via regularization in the motion field solution. In this case a pairwise smoothness term should be utilized:

$$\lambda^{(ij)}_{s} = \lambda_w b_{ij} \times \begin{cases} exp\left(-\left|J^{(i)}_{d,ovr}(j) - J^{(j)}_{d,ovr}(i)\right| / b_{ij}\right) & |\Omega_{ij}| > 0 \\ \\ 1/\left(1 + exp\left(\rho^{(ij)}_{u}(0) - \rho^{(ij)}_{r}(\zeta)\right)\right) & o.w. \end{cases} \tag{3.46}$$

in which the data cost difference is normalized with the common boundary length, rather than the number of overlapping pixels, making the same layer decision more probable for large number of overlapping pixels and less probable for a small number of overlapping pixels. If two SPs are overlapping, and the data cost difference in overlapping region large, then the pairwise smoothness term would approach to zero, allowing neighboring SPs to move independently. If the data cost difference in the overlapping region is small, then SPs are forced to move together, since the overlapping might be due to the errors in the motion estimates. For non-overlapping

case, for similar SPs independent layer assignment penalty, $\rho_r(\zeta)$, is large; therefore, in order to have a small smoothness term the difference between the motion vectors should be larger. However, if neighboring pixels are not similar, then the layer order penalty will be close to zero, which allows to have smaller smoothness terms, even for the small differences of the motion vectors. Since the relation between λ_s and λ_r affects the layer order selection, to control the overall smoothness a new weight, λ_w is utilized rather than the smoothness weight λ_s.

Regularized Lucas-Kanade Solution for Motion Field

Given the visible region of each SP, the pairwise smoothness weights and the initial estimate of motion vectors ($\{\hat{u}_i\}_{i=1}^{N}$), motion vector of SP i should be updated such that the objective function (3.40) is minimized. Getting the part of the objective function related with the error in initial estimate of SP i, the following minimization must be performed to obtain the required update:

$$\hat{\delta}_i = \arg\min_{\delta_i} \sum_{x \in \hat{\Omega}_{i,k-1}^{(v)}} \sum_{c} \frac{\left(I_{c,k}(x+\hat{u}_i) - I_{c,k-1}(x-\delta_i)\right)^2}{\sigma_{c,i}^2} + 2\sum_{j \in \Gamma_i} \lambda_s^{(ij)} ||\hat{u}_i + \delta_i - \hat{u}_j||^2 \tag{3.47}$$

where δ_i is the error of the initial estimate. The normalization of each channel with its variance in (3.47) is proposed to gain robustness against small registration errors in overall cost function. However, if each channel is normalized with its variance, then the channels having larger variances would become less significant in LK solution, while they usually contain more significant gradient information, which is essential in LK solution. In order to overcome this problem, in LK solution, rather than the individual variances of the channels, all channels of a SP are normalized with the average of the variances over channels, and the minimization problem becomes following:

$$\hat{\delta}_i = \arg\min_{\delta_i} \frac{1}{\sigma_i^2} \sum_{x \in \hat{\Omega}_{i,k-1}^{(v)}} \|I_k(x+\hat{u}_i) - I_{k-1}(x-\delta_i)\|^2 + 2\sum_{j \in \Gamma_i} \lambda_s^{(ij)} ||\hat{u}_i + \delta_i - \hat{u}_j||^2 \tag{3.48}$$

where

$$\sigma_i^2 = \frac{1}{C} \sum_{c=1}^{C} \sigma_{c,i}^2 \tag{3.49}$$

The minimization problem defined in (3.48) is equivalent to :

$$\hat{\delta}_i = \arg\min_{\delta_i} \sum_{x \in \hat{\Omega}_{i,k-1}^{(v)}} ||I_k(x + \hat{u}_i) - I_{k-1}(x - \delta_i)||^2 + \lambda_i ||\delta_i - \bar{\delta}_i||^2 \tag{3.50}$$

where λ_i and $\bar{\delta}_i$ are defined as:

$$\lambda_i = 2\sigma_i^2 \sum_{j \in \Gamma_i} \lambda_s^{(ij)} \tag{3.51}$$

$$\bar{\delta}_i = -\hat{u}_i + \frac{1}{\sum_{j \in \Gamma_i} \lambda_s^{(ij)}} \sum_{j \in \Gamma_i} \lambda_s^{(ij)} \hat{u}_j \tag{3.52}$$

Note that $\bar{\delta}_i$ is the robust estimate of the update vector due to the neighboring motion vectors; the individual weights, $\lambda_s^{(ij)}$, for the neighboring motion vectors, $\hat{u}_j$, are adaptively determined based on common boundary length, b_{ij}, SP similarity s_{ij} and the relative data cost on overlapping region, $J_{d,ovr}^{(i)}(j)$. Applying Taylor series expansion to I_{k-1} on (3.50), taking derivative with respect to δ_i and equating to zero as in (3.11), update for the initial estimate is obtained:

$$\hat{\delta}_i = \left(A_{i,k-1}^{(v)} + \lambda_i I_{2\times 2} \right)^{-1} \left(b_{i,k-1}^{(v)}(\hat{u}_i) + \lambda_i \bar{\delta}_i \right) \tag{3.53}$$

where $A_{i,k-1}^{(v)}$ and $b_{i,k-1}^{(v)}(\hat{u}_i)$ are the structure tensor and the error vector obtained from the visible part of SP i:

$$A_{i,k-1}^{(v)} = \sum_{x \in \Omega_{i,k-1}^{(v)}} \begin{bmatrix} ||I_h(x)||^2 & I_h^T(x) I_v(x) \\ I_h^T(x) I_v(x) & ||I_v(x)||^2 \end{bmatrix} \tag{3.54}$$

$$b_{i,k-1}^{(v)}(\hat{u}_i) = \sum_{x \in \Omega_{i,k-1}^{(v)}} \begin{bmatrix} I_h^T(x) \left(I_{k-1}(x) - I_k(x + \hat{u}_i) \right) \\ I_v^T(x) \left(I_{k-1}(x) - I_k(x + \hat{u}_i) \right) \end{bmatrix} \tag{3.55}$$

The solution given in (3.53) is a regularized version of classical LK method [26]. This solution is also quite similar to the classical HS approach [25], but the region of support for the motion vector is larger (i.e. the visible part of the SP), and the expected value of the motion vector defined in a novel way.

Using the result of (3.53), motion vectors are updated for the next iteration:

$$\hat{u}_i^{(n+1)} = \hat{u}_i^{(n)} + \hat{\delta}_i \tag{3.56}$$

Mean-Shift Solution for Motion Field

The solution given in (3.53) is a valid solution, if the error in initial estimate, δ_i, is small with respect to the spatial derivative step size, which might not be the case. Since the motion of the SPs is assumed to be translational and SPs are assumed to be rigid, the motion vector should be equal to the displacement of the SP centroid.

The spectral distribution of an SP can be considered as the parameters of a Gaussian random process, which generates the pixels in an SP. In this case, if the brightness consistency assumption holds, then the pixels in an SP in the previous and current frames should be generated by the same random process. Assuming that a single SP will be placed on the current frame and a good initial estimate is available, that is the difference between $\hat{\boldsymbol{u}}_i^{(0)}$ and $\boldsymbol{u}_i$ is small with respect to the SP size, then moving the SP center to its expected position in each step, the location where the observation probability is maximized can be reached. Considering the maximum of the likelihood function as the mode of a probability distribution function, this approach corresponds to the mean-shift algorithm [43] in the spatial domain where the kernel is defined by SP membership function and spectral distribution. If it was guaranteed that the error of initial estimates are small enough, the SPs will not occlude, and the motion of every SP is observable (no aperture effect on SP level), then the mean-shift approach would converge to the true global solution. Under these assumptions, placing the SPs of the previous frame on the current frame with initial displacement estimates and moving the SPs towards their centroids, which is computed by using the spectral characteristics obtained in the previous frame, the displacements of SPs can be obtained iteratively.

For the visible region of SP, $\Omega_{i,k-1}^{(v)}$, the centroid of the SP on frame t with the displacement vector u is defined as (3.57):

$$\bar{\boldsymbol{x}}_{i,t}(u) = \frac{\sum_{x \in \Omega_{i,k-1}^{(v)}} (x+u)\, p\left(I_t(x+u) | \mu_{i,k-1}, \Sigma_{i,k-1}\right)}{\sum_{x \in \Omega_{i,k-1}^{(v)}} p\left(I_t(x+u) | \mu_{i,k-1}, \Sigma_{i,k-1}\right)} \tag{3.57}$$

where $(\mu_{i,k-1}, \Sigma_{i,k-1})$ are the parameters of Gaussian random process of SP i obtained in the previous frame. The difference between the centroids on the previous and on the current frames provides an estimate for the motion vector update:

$$\hat{\delta}_i = \bar{\boldsymbol{x}}_{i,k}(\hat{u}_i) - (\bar{\boldsymbol{x}}_{i,k-1}(0) + \hat{u}_i) \tag{3.58}$$

Once the estimates of the motion vector updates are obtained, the motion vectors for the next iteration should be smoothed due to the regularization term:

$$\hat{u}_i^{(n+1)} = \hat{u}_i^{(n)} + \frac{1}{w_i + \lambda_i}\left(w_i\hat{\delta}_i + \lambda_i\bar{\delta}_i\right) \tag{3.59}$$

where w_i is the self weight, λ_i and $\bar{\delta}_i$ are the smoothness weight and the expected update defined as in (3.51) and (3.52) respectively. The self weight is obtained by multiplying the total boundary length of the SP with total spectral variance:

$$w_i = \sigma_i^2 \sum_{j \in \Gamma_i} b_{ij} \tag{3.60}$$

Even if the mean-shift solution does not result in precise sub-pixel motion estimates, for large displacements, it improves correcting the initial estimate and makes it possible to converge to the true solution with regularized LK solution.

Motion Estimation Quality based Adaptation

Both MS and regularized LK solutions are the gradient descent solutions which require good initial estimates for convergence [1, 26, 43]. Since the algorithm works with a single initial estimate for each SP, it is crucial to eliminate erroneous initial estimates. Moreover, the erroneous estimates might mislead their neighbor SPs through the smoothness term. In order to determine the quality of the initial estimates, the data penalty in (3.30) should be utilized. The match score of a SP in the visible region of support is measured by the following expression:

$$\eta_i^{(v)} = \frac{1}{|\Omega_{i,k-1}^{(v)}|} \sum_{x \in \hat{\Omega}_{i,k-1}^{(v)}} exp\left(-\sum_c \frac{(I_{c,k}(x+\hat{u}_i) - I_{c,k-1}(x))^2}{2\sigma_{c,i}^2}\right) \tag{3.61}$$

where c are the image color channels and $\sigma_{c,i}^2$ is the variance of c^{th} channel for SP i. The score in (3.61) corresponds to average pixel observation probability when the measurement noise is independent, Gaussian with variance equal to the variance of SP. Once the match scores are obtained, the mismatched set is obtained by thresholding match scores:

$$W^{(n)} = \{i | \eta_i^{(v)} < \eta_{th}\} \tag{3.62}$$

Using the match scores as an additional weight and discarding the mismatched SPs, LK and MS solutions become more reliable. For this purpose, the relations in (3.51)

and (3.52) are extended to incorporate the match score, $\eta_j^{(v)}$:

$$\lambda_i = 2 \sum_c \sigma_{c,i}^2 \sum_{j \in \Gamma_i / W} \eta_j^{(v)} \lambda_s^{(ij)} \tag{3.63}$$

$$\bar{\delta}_i = \frac{1}{\sum_{j \in \Gamma_i / W} \eta_j^{(v)} \lambda_s^{(ij)}} \sum_{j \in \Gamma_i / W} \eta_j^{(v)} \lambda_s^{(ij)} (\hat{u}_j - \hat{u}_i) \tag{3.64}$$

With the match score weight, LK and MS update equations (3.53, 3.59) can be expressed as (3.65) and (3.66), respectively.

$$\hat{\delta}_i = \left(\eta_i^{(v)} A_{i,k-1}^{(v)} + \lambda_i I_{2\times 2} \right)^{-1} \left(\eta_i^{(v)} b_{i,k-1}^{(v)} (\hat{u}_i) + \lambda_i \bar{\delta}_i \right) \tag{3.65}$$

$$\hat{u}_i^{(n+1)} = \hat{u}_i^{(n)} + \frac{1}{\eta_i^{(v)} w_i + \lambda_i} \left(\eta_i^{(v)} w_i \hat{\delta}_i + \lambda_i \bar{\delta}_i \right) \tag{3.66}$$

Once the solution for the SPs, except the mismatched ones, are obtained, the motion vectors of the mismatched SPs are initiated with the neighbor SP motion estimate which results in minimum data penalty:

$$\hat{u}_i^{(n+1)} = \arg \min_{u \in U_i^{(n)}} \sum_{x \in \Omega_{i,k-1}} \rho_d (I_k(x+u) - I_{k-1}(x)) \tag{3.67}$$

where $U_i^{(n)} = \{ \hat{u}_j^{(n+1)} | j \in \Gamma_i / W^{(n)} \}$. Replacing the erroneous motion estimates with the motion estimates with neighbors also employed in [8] and quite similar to 3DRS [38].

Summary of the Proposed ICM-based Solution

ICM solution is a computationally efficient algorithm for the SP based motion estimation problem defined in (3.41). In this approach, given the motion estimates the occlusion regions and pairwise smoothness weights are determined, and excluding the occlusion regions, the motion estimates are updated using pairwise smoothness weights. Performing these two steps iteratively the MAP estimates for the layer orders and the motion field are obtained. Key points in this approach are updating the visible region of support, $\Omega_{i,k-1}^{(v)}$, and pairwise smoothness weights, $\lambda_s^{(ij)}$, on the layer order solution step; weighting the motion vectors with their reliability, η_i, and pairwise smoothness weights on the motion estimation step, and replacing the unreliable motion vectors with the motion vectors of the neighbor SPs. ICM solution is summarized in Table 3.1.

Table 3.1: ICM solution pseudo-code

1. *Initiate motion vectors:* $\hat{u}_i^{(0)} = [0\ 0]^T\ \forall\ i$,
2. *Given the motion estimates* $\hat{u}_i^{(n)}$*:*
   i. *Obtain the robust layer order estimates,* $\breve{R}_{ij}$ *using (3.45)*
   ii. *Obtain the visible region of support for SPs,* $\hat{\Omega}_{i,k-1}^{(v)}$ *using (3.26)*
   iii. *Obtain the pairwise smoothness weight,* $\lambda_s(ij)$ *using (3.46)*
3. *Given the region of supports,* $\hat{\Omega}_{i,k-1}^{(v)}$ *and the smoothness weights,* $\lambda_s(ij)$*:*
   i. *Obtain the match scores,* $\eta_i^{(n)}$*, for the current estimates,* $\hat{u}_i^{(n)}$*, using (3.61),*
   ii. *Determine the set of mismatched SPs,* $W^{(n)}$ *using (3.62),*
   iii. *Update the motion vectors,* $\hat{u}_i^{(n+1)}$ *using (3.65) or (3.66),*
   v. *Initiate the motion vectors of the mismatched SPs with the neighbor SP motion vectors using (3.67),*
4. *Return to the Step-2 and continue until convergence.*

3.2.2.2 Particle Belief Propagation Solution

As explained in the ICM based solution, given a pair of neighboring SP motion vectors, the layer order can be uniquely determined by minimizing the sum of the layer prior penalty (3.33) and the motion prior penalty (3.36). If a set of candidate motion vectors are given for a particular SP, for each neighboring SP motion vector pair, the layer order can be determined; the layer prior and motion prior penalty can also be obtained easily. Using the selected layer order, data cost for each SP can be obtained. Hence, the motion vector pair minimizing the joint cost function of the SP pair can be obtained. If the objective were to minimize the joint cost function of a pair of SPs, then following these few steps, the solution can be achieved. However, since the selected motion vector of a particular SP also changes the cost for its neighboring SPs, the pairwise solution would not result in the minimum energy solution. In order to select the motion vectors from the given candidate set which minimizes the overall energy function, a global optimization method should be utilized.

Overview of Particle Belief Propagation

Particle belief propagation [50] is quite similar to the belief propagation [47], except the messages are defined on particles, not on whole set. For the SP based motion estimation problem, defining the nodes as SPs and the edges as the connection between

SPs, loopy particle belief propagation can be applied. In Figure 3.6, nodes (i, j, k), edges (connection between the nodes), particles $\{...pqr...\}$ and message from SP i to particle q of SP j is illustrated.

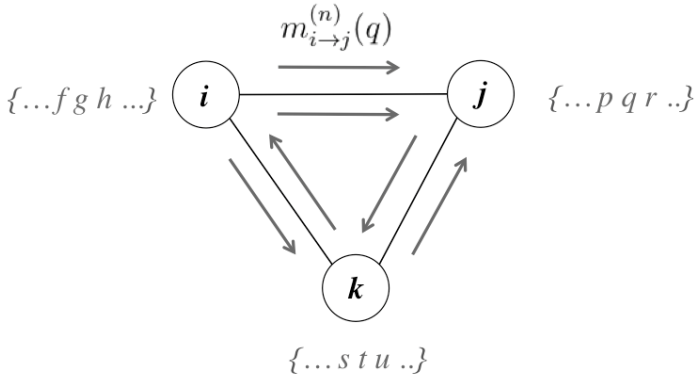

Figure 3.6: Belief propagation message passing

The message from SP i to particle q of SP j is defined as the sum over particles of i weighted by the neighbor beliefs:

$$m_{i\to j}^{(n)}(q) = \frac{1}{N_{ij}} \sum_{h=1}^{H_i} \Phi_i(h)\Psi_{ij}(h,q) \prod_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \tag{3.68}$$

where H_i is the number of particles for SP i and N_{ij} is the normalization factor for the messages form i to j:

$$N_{ij} = \sum_{q=1}^{H_j} \sum_{h=1}^{H_i} \Phi_i(h)\Psi_{ij}(h,q) \prod_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \tag{3.69}$$

The message passing algorithm described in (3.68) is utilized for estimating the marginals and also known as sum-product algorithm [51]. As an alternative, max-product algorithm is utilized for maximum-likelihood estimation [51]. The messages in max-product algorithm are defined as (3.70):

$$m_{i\to j}^{(n)}(q) = \frac{1}{N_{ij}} \max_h \left(\Phi_i(h)\Psi_{ij}(h,q) \prod_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \right) \tag{3.70}$$

When performed in log domain, products in the max-product algorithm are replaced with summations (3.71) which is called as the max-sum algorithm [51]:

$$m_{i\to j}^{(n)}(q) = \max_h \left(\Phi_i(h) + \Psi_{ij}(h,q) + \sum_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \right) - \log N_{ij} \tag{3.71}$$

Although $\Phi(.)$ and $\Psi(.)$ stand for probability-like functions in (3.68) and (3.70), with a slight abuse of notation, in (3.71) they stand for the negative penalty functions. Since the problem defined in (3.41) is a MAP estimate problem, the max-sum algorithm can be utilized. For the positive penalty functions, messages can be obtained by minimizing the cost over particles:

$$m_{i\to j}^{(n)}(q) = \min_{h} \left(\Phi_i(h) + \Psi_{ij}(h,q) + \sum_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \right) + \log N_{ij} \tag{3.72}$$

Once the messages are obtained, the neighbor beliefs can be defined and the maximum-likelihood (ML) estimate can be obtained as:

$$\hat{q} = \arg\min_{q} \left(\Phi_j(q) + \sum_{i\in\Gamma_j} m_{i\to j}^{(n)}(q) \right) \tag{3.73}$$

Belief Propagation for SP-based Motion Estimation

In the proposed belief propagation (BP) solution, a particle contains only the motion vector of mentioned SP, but not the depth relative to the neighbor SPs and neighbor SP motion vectors. For the problem defined in (3.41), the functions in (3.72) are:

$$\Phi_i(h) = \sum_{x\in\Omega_{h,i,k-1}^{(v)}} \rho_d \left(I_k(x + u_i^{(h)}) - I_{k-1}(x) \right) + |\Omega_{h,i,k-1}^{(o)}| \tag{3.74}$$

$$\Psi_{ij}(h,q) = \rho_u^{(ij)} \left(R_{ij}^{(hq)} \right) + \rho_r^{(ij)} \left(R_{ij}^{(hq)} \right) \tag{3.75}$$

where $u_i^{(h)}$ is the motion vector for the particle h of SP i, $\rho_u^{(ij)}(.)$ is the motion prior penalty defined in (3.36), and $\rho_r^{(ij)}(.)$ is the layer prior penalty defined in (3.33) for the layer order of particles h and q. While evaluating (3.72), the last term of the expression is the neighbor beliefs which already obtained in the previous iteration. The regularization cost is determined by the selected layer order. Given a particle pair, the layer order can be selected by minimizing the regularization cost as expressed in (3.44), in which the occluding one can be obtained using the data cost in the overlapping region. In other words, given a pair of motion vectors, there is no need to generate different particles for different layer orders, since the minimum cost layer order is clearly defined. However, the visible and occluded pixels in (3.74) depends on not only the motion vector of SP i, but also the motion vectors of the neighboring SPs. The data cost for hypothesis h of SP i without overlapping is defined as:

$$\phi_i(h) = \sum_{x \in \Omega_{i,k-1}} \rho_d(I_k(x + u_i^{(h)}) - I_{k-1}(x)) \tag{3.76}$$

If the particle h is paired with the particle q of SP j, then the overlapping region would yield to an additional cost, if SP j is occluding SP i for the particles h and q:

$$\phi_{ij}(h,q) = \delta\left[R_{ij,hq} + 1\right] \sum_{x + \tilde{u}_i^{(h)} \in \Omega_{ij,hq}} \left(\lambda_o - \rho_d\left(I_k(x + u_i^{(h)}) - I_{k-1}(x)\right)\right) \tag{3.77}$$

where $\Omega_{ij,hq}$ is the set of overlapping pixels and $R_{ij,hq}$ is the layer order for the particles h and q. (3.77) states that, if SP j is occluding SP i for the particles h and q, the data cost should be replaced with occlusion cost in the overlapping region. If the case of multiple overlapping neighbors for a single pixel is discarded, then the data cost function can be expressed as a function of the motion vector of the particle h and the selected neighbor particles:

$$\Phi_i(h) = \phi_i(h) + \sum_{j \in \Gamma_i} \phi_{ij}(h, q_j)\,(h, q_j) \tag{3.78}$$

where q_j is the selected particle for SP j. While computing the message from i to the particle q of SP j, since the message is computed for q, the data term of SP i should be evaluated for the particle q of SP j. For the other neighbors, the particles selected in previous iteration can be utilized. Since the purpose of the max-sum algorithm is to obtain the ML estimate, and the data cost in (3.73) depends on the selected neighbor particles; first selecting the neighbor particle h maximizing the belief (or minimizing the prior cost) for each hypothesis of SP j with (3.72), then selecting the ML particle for SP j with (3.73) may not result in ML estimate. Therefore, the message sending particle (or matching particle) of SP i for the hypothesis q of SP j should be obtained by minimizing the cost for posterior:

$$h_{i \to (q,j)} = \min_h \left(\Phi_j(q) + \Phi_i(h) + \Psi_{ij}(h,q) + \sum_{z \in \Gamma_i / j} m_{z \to i}^{(n-1)}(h) \right) \tag{3.79}$$

Minimization of (3.79) corresponds to selecting the hypothesis of SP i which maximizes the posterior for the hypothesis q of SP j. Let $\Phi_i^{(n)}(h)$ denote the data cost for the hypothesis h of SP i at n^{th} iteration, and $h_{i \to (q,j)}^{(n)}$ be the matching particle of SP i for the particle q of SP j, then the minimization problem (3.79) for matching particle can be expressed as (3.80):

$$h_{i\to(q,j)}^{(n)} = \min_{h} \begin{pmatrix} \Phi_j^{(n-1)}(q) + \phi_{ji}(q,h) - \phi_{ji}(q, z_{i\to(q,j)}^{(n-1)}) + \\ \Phi_i^{(n-1)}(h) + \phi_{ij}(h,q) - \phi_{ij}(h, z_{j\to(h,i)}^{(n-1)}) + \\ \Psi_{ij}(h,q) + \sum_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \end{pmatrix} \tag{3.80}$$

Since $\Phi_j^{(n-1)}(q)$ and $\Phi_i^{(n-1)}(h)$ are constant, they can be discarded, and the matching particle can be selected by considering the overlapping region cost, the regularization cost and the beliefs from the previous iteration. Once the matching particles are selected, messages can be generated:

$$m_{i\to j}^{(n)}(q) = \left(\begin{array}{r} \Phi_i^{(n-1)}(h) + \phi_{ij}(h,q) - \phi_{ij}(h, z_{j\to(h,i)}^{(n-1)}) \\ +\Psi_{ij}(h,q) + \sum_{z\in\Gamma_i/j} m_{z\to i}^{(n-1)}(h) \end{array} \right) \Bigg|_{h=h_{i\to(q,j)}^{(n)}} \tag{3.81}$$

For the first iteration $(n = 1)$, matching particles can be obtained by minimizing the binary cost:

$$h_{i\to(q,j)}^{(1)} = \arg\min_{h} \left(\phi_{ji}(q,h) + \phi_{ij}(h,q) + \Psi_{ij}(h,q)\right) \tag{3.82}$$

Using these initial matching particles the data cost can be initiated using (3.78). Given the initial data cost and the matching particles, BP iterations can be started. Using the matching particles, the messages can be obtained using (3.81), and when a BP iteration is completed, the matching particles are updated according to (3.80). Belief propagation message passing algorithm is summarized on Table 3.2.

Birth Process for Generating New Particles

In order to obtain a computationally efficient algorithm with the ability of determining global minimum or at least ability to converge to a local minimum in a large neighborhood, the most critical step is the particle generation. In [42], starting from a dense motion field solution, on each iteration, particles are generated in the neighborhood of the current estimate. Rather than requiring an initial dense or sparse motion field solution, or randomly sampling the possible locations for each SP, the proposed ICM-based solution can be utilized for particle generation. When the initial estimate is close to the global minimum, then the ICM-based solution would converge for those SPs. However, if a SP is severely occluded, then the ICM-based solution might

Table 3.2: Belief propagation iterations pseudo-code

1. *Given the particles, for each neighbor particle pair* (h, q)*:*
   i. *Compute the overlapping region cost,* $\phi_{ij}(h, q)$ *using (3.77),*
   ii. *Determine layer order,* $R_{ij,hq}$ *and compute regularization cost* $\Psi_{ij}(h, q)$ *using (3.44) and (3.33, 3.36),*
2. *For each particle compute the occlusion free data cost,* $\phi_i(h)$ *using (3.76)*
3. *Initiate the matching particles,* $h^{(1)}_{i\rightarrow(q,j)}$ *using (3.82),*
4. *Initiate the data cost for each particle,* $\Phi^{(0)}_i(h)$ *using the matching particles* $h^{(1)}_{i\rightarrow(q,j)}$ *and (3.78),*
5. *For the maximum number of iterations:*
   i. *Compute the messages,* $m^{(n)}_{i\rightarrow j}(q)$ *using (3.81),*
   ii. *Update the matching particles for the next iteration,* $h^{(n+1)}_{i\rightarrow(q,j)}$ *using (3.80),*
6. *For each SP select the ML particle,* $\hat{h}_i$ *using (3.73).*

not able to find the solution, or even worse, those occluded SPs tend to move into the uniform regions to avoid the occlusions. The moving away alternative for the layer order might also be a problem, since it has a marginal cost for the SPs with different colors. If the displacement of an SP is large with respect to SP size and if that SP is moving independent from its neighbors, then it might not be possible to find the true solution either. Birth process should be able to compensate those problems related to the ICM-based solution.

At the output of the ICM-based solution, if two SPs are moving away from each other, then an alternative hypothesis, moving together assumption might be useful. If two SPs are overlapping at the output, this might be either due to the occlusion, or erroneous motion estimate of one of them. If observations do not significantly support one of the SPs as the occluding one (data cost difference in the overlapping region is large) then new particles might help to explain the observations. Utilization of data cost difference in overlapping region would also help to prevent generating unnecessary particles, since it is expected to be large for true occlusions. If an SP has a low match score even for the visible region, or it has a occluded solution although it is similar to none of its neighbors, then randomly sampling the possible locations around the initial estimate would help.

Given the motion vector $u_j^{(q)}$ of the neighbor SP j, a new particle might be generated for SP i, if no particle with a similar motion vector exist in the set of SP i. In this case, the first step is to select the nearest particle to the given motion vector:

$$h = \arg\min_z \|u_i^{(z)} - u_j^{(q)}\| \tag{3.83}$$

Using the nearest particle, probability of q not being the particle set of i can be approximated with (3.84):

$$P_{(q,j)/i} = 1 - exp(-\|u_i^{(h)} - u_j^{(q)}\|^2) \tag{3.84}$$

In order to make SP i to generate a new particle, the match score for the neighbor particle should be high, the neighbor's particle should not be in the set of SP i and if two SPs are overlapping, then the data cost difference should be large. Using these constraints, the birth probability can be defined as:

$$P_{birth,i}(q,j) = \begin{cases} 0 & \tilde{u}_i^{(h)} = \tilde{u}_j^{(q)} \\ \eta_j(q) P_{(q,j)/i} exp\left(-\frac{\left|J_{d,ovr}^{(h,i)}(q,j) - J_{d,ovr}^{(q,j)}(h,i)\right|}{\Omega_{ij,hq}}\right) & |\Omega_{ij,hq}| > 0 \\ \eta_j(q) P_{(q,j)/i} & o.w. \end{cases} \tag{3.85}$$

where $\Omega_{ij,hq}$ is the overlapping region and $\eta_j(q)$ is the match score for the hypothesis q of SP j computed on whole region of support:

$$\eta_j(q) = exp\left(-\frac{1}{|\Omega_{j,k-1}|} \sum_{x \in \Omega_{j,k-1}} \sum_c \frac{(I_{c,k}(x + u_j^{(q)}) - I_{c,k-1}(x))^2}{\sigma_{c,j}^2}\right) \tag{3.86}$$

In (3.85), if there exist an integer casted motion vector equal to the neighbor, then this particle is discarded since a particle with the same region of support already exist. Once the birth probability is obtained, new particles can be generated randomly by thresholding this value with a random number.

Belief Propagation Solution Summary

Starting with zero initial estimates, at the first iteration ICM-based solution is performed. Then the new particles are generated through the birth process, based on the results of the ICM-based solution. Following the BP iterations, the matching particles are determined. Using the matching particles instead of neighbor motion vectors in the ICM-based solution and their motion vectors as the initial estimates, iterations continue with the ICM-based solution. At the end, the ML particle for each SP selected. BP solution is summarized in Table 3.3.

Table 3.3: Belief propagation solution pseudo-code

1. *Initiate the motion vectors of the particles:* $u_i^{(h)} = [0\ 0]^T\ \forall\ i$*, and the matching particles* $h_{i\rightarrow(q,j)} = 1\ \forall\ (i,j)\ st.\ b_{ij} > 0$*,*
2. *Perform the steps [2-4] of ICM-based solution given in Table 3.1 with the given matching particles,*
3. *For* N_{birth} *iterations:*
   i. *For each neighbor particle determine the birth probability,* $P_{birth}^{(ij)}(q)$ *using (3.85),*
   ii. *If* $P_{birth}^{(ij)}(q) > th$*, generate a new particle for SP* i *with the motion vector* $u_j^{(q)}$*,*
4. *Perform the steps [1-5] of BP iteration as described in Table 3.2*
5. *Return to the Step-2 and continue until convergence,*
6. *For each SP select the ML particle,* $\hat{h}_i$ *using (3.73).*

3.2.3 Hierarchical Superpixels and Pyramidal Motion Estimation

Hierarchical methods are widely utilized for motion estimation; since the large displacements can be easily solved in lower resolutions and coarse-to-fine strategy reduces the computational complexity. In the proposed motion estimation method, displacements are assumed to be small with respect to SP size, therefore larger SPs are needed to handle larger displacements. On the other hand, representative power of SPs increases with the decreasing cluster size, therefore smaller SPs should be preferred (for the limiting case cluster size is equal to pixel size which results in the exact representation of the image). In lower resolutions the details are lost; therefore, the larger SPs may still provide enough representative power in those resolutions. Since the SPs in full resolution are given, rather than finding the SPs in lower resolutions independently, the SPs in higher resolution may be grouped to obtain larger SPs. Such grouping approach is known as hierarchical agglomerative clustering [52] and commonly employed in computer vision problems.

Hierarchical Superpixels

In order to have visible SPs in the lower resolutions, SPs should have a certain size. If the squared perimeter/area ratio of a SP is large, when it is downsampled, most of its pixels will be blended with other SPs which is not preferred. These preferences are the spatial constraints for merging. On the other hand, in order to merge two

SPs, they should have similar appearances, which is a strong indication that those SPs might belong to the same object. The visual dissimilarity of the SPs is utilized as the spectral cost for merging and defined as:

$$J_v(i,j) = (\mu_i - \mu_j)^T (\Sigma_i + \Sigma_j)^{-1} (\mu_i - \mu_j) \tag{3.87}$$

Due to the spatial constraints, keeping a cluster as it is but not merging with other clusters will have a cost, which is utilized as the spatial cost:

$$J_s(i) = \lambda_1 \left(1 - \frac{a_i}{\bar{a}}\right) + \lambda_2 \left(\frac{p_i^2}{4\pi a_i} - 1\right) \tag{3.88}$$

where $\bar{a}$ is the expected area of a SP in given resolution, a_i is the area of SP i, p_i is the perimeter of SP i (number of boundary pixels) and λ_1 and λ_2 are corresponding weights. While keeping SPs as they are has a cost, merging a SP with another SP will also have a cost. The cost defined in (3.88) is also valid for the merged SPs, but for the resulting merged SPs area and perimeter:

$$J_s(i,j) = \lambda_1 \left(1 - \frac{a_{ij}}{\bar{a}}\right) + \lambda_2 \left(\frac{p_{ij}^2}{4\pi a_{ij}} - 1\right) \tag{3.89}$$

where a_{ij} is the area and p_{ij} is the perimeter after merging. When a SP is merged with another SP, the cost of keeping distinct SPs (3.88) will disappear and cost of the merging will appear (3.87),(3.89), and the total cost of merging is obtained as:

$$J(i,j) = J_v(i,j) + J_s(i,j) - J_s(i) - J_s(j) \tag{3.90}$$

Starting from the smallest cost, the SPs are merged iteratively, until the required number of SPs is reached. Merging the SPs having quite different spectral distributions might not be preferred. In such cases, merging can be stopped, if the spectral dissimilarity is larger than a given threshold.

A sample output of this method is presented in Figure 3.7. As it is shown in the figure, the method results in regularly shaped larger clusters while it still preserves the local structure in the image.

Pyramidal Motion Estimation

Pyramidal solution is applicable for the proposed motion estimation method. Given the SPs in the original resolution, hierarchical SPs can be obtained iteratively by minimizing (3.90). For each level of a P level pyramid, required number of SPs is set to

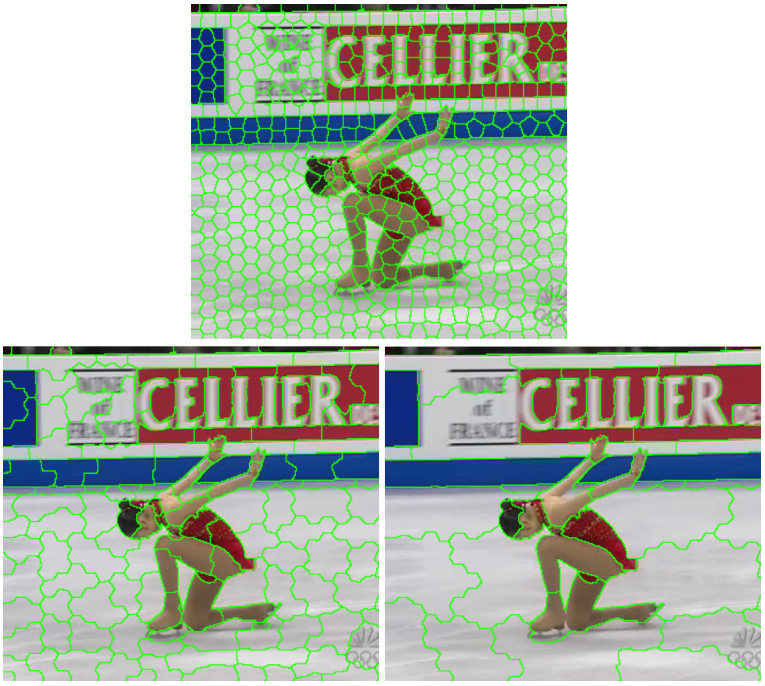

Figure 3.7: Hierarchical superpixels.

25% of the number of SPs in higher resolution. Then both the SP masks and the original image are down-sampled by two, to obtain a lower resolution, which is needed to reduce the computational the complexity and keep the representational power of SPs. This approach helps to keep the average SP area constant and get the similarly sized SPs on each resolution which are able to handle approximately same displacement in the resolution they are defined, but larger displacements in higher resolutions. Starting from the lowest resolution, on each pyramid level motion estimation is performed. As algorithm moves to a higher resolution, motion vector of each SP is initiated with the estimate of its parent in the previous resolution. The pyramidal extension of the proposed algorithm is summarized in Table 3.4 as a pseudo-code.

3.3 Experiments

In the first part of this section, the performance metrics for optical flow solution are presented, which are defined in [6]. In order to understand the proposed method better, performance of the proposed alternatives are measured by using these metrics. Then the selected solution is compared against the state-of-the-art methods using the metrics and database defined in [6].

Table 3.4: Pyramidal solution pseudo-code

1. *Perform SP merging by minimizing (3.90) for P levels to get larger clusters,*
2. *From the lowest resolution to the highest resolution,*
   i. *Get the SPs for the selected resolution, downsample the original image and SP masks for the selected level,*
   ii. *Initiate the motion vectors of the SPs with the estimate of its parents in the previous resolution, use zero initial estimate for the lowest resolution,*
   iii. *Apply the selected solution.*

3.3.1 Performance Measures for Optical Flow Field

Performance of the optical flow methods is measured with *angular error, endpoint error, interpolation error* and *normalized interpolation error* [6]. Angular error is defined as the angle between the ground-truth motion vector and the estimated motion vector in normalized coordinates (3.91):

$$AE = \cos^{-1} \frac{< \boldsymbol{u}, \boldsymbol{u}_{gt} >}{\|\boldsymbol{u}\| \, \|\boldsymbol{u}_{gt}\|} \tag{3.91}$$

where $\boldsymbol{u} = [u_x \;\; u_y \;\; 1]^T$ in which u_x and u_y stand for the horizontal and vertical displacements.

Endpoint error is defined as the L2 distance between the ground-truth motion vector and the estimated motion vector (3.92):

$$EE = \|\boldsymbol{u} - \boldsymbol{u}_{gt}\| \tag{3.92}$$

Interpolation error is defined as the average L2 distance between the interpolated image and the ground-truth image (3.93):

$$IE = \left(\frac{1}{N} \sum_{x} \|I_{k+1}(\boldsymbol{x}) - I_k(\boldsymbol{x} - \boldsymbol{u})\|^2 \right)^{1/2} \tag{3.93}$$

Normalized interpolation error is defined similar to the interpolation error, but an additional normalization term depending on the magnitude of the image gradients is also included (3.94):

$$IE = \left(\frac{1}{N} \sum_{x} \frac{\|I_{k+1}(\boldsymbol{x}) - I_k(\boldsymbol{x} - \boldsymbol{u})\|^2}{\|\nabla I_{k+1}(\boldsymbol{x})\| + \epsilon} \right)^{1/2} \tag{3.94}$$

In Middlebury evaluation database [6], the performances of various optical flow methods are presented. In addition to individual statistics of four error types mentioned above, for each error type, an average rank for each method is also reported.

3.3.2 Performance of Proposed Alternatives

Proposed alternatives are evaluated with the test data in Middlebury database [6]. For the motion field solution mean-shift and regularized Lucas-Kanade approaches, and for the proposed optimization iterated conditional modes and belief propagation based methods are considered. The following alternatives are evaluated:

- Mean-shift ICM solution (MS-ICM)
- Regularized Lucas-Kanade ICM solution (LK-ICM)
- Regularized Lucas-Kanade belief propagation solution (LK-BP)

For all alternatives, the pyramidal solution is applied for which a pyramid with three levels is utilized.

In order to understand the effect of the regularization term on the solution, regularization alternatives are compared against each other. Comparison of regularization alternatives is performed by using the LK-ICM approach. As the first alternative, expected update vector, $\bar{\delta}_i$, due to neighbor vectors is set to zero, and a regularization term added to structure tensor to handle the SPs having poor-conditioned structure tensors. As the second alternative common boundary length utilized as the neighbor weights. SP similarity and move together probability weighted common boundary length are utilized as the neighbor weight for the third and the fourth alternatives, respectively. As the last alternative, neighbor weights are determined by multiplication of match score, move together probability and common boundary length. For each alternative λ_w is optimized to minimize the total end-point-error in Middlebury test data. For the last two alternatives, λ_w is used in the equation (3.46) to obtain $\lambda_s^{(ij)}$. The utilized regularization terms and the sum of corresponding endpoint errors on eight test images are presented in Table 3.5. The term σ_i^2 shown in the first column is the average of variances over image channels for SP i as utilized in (3.51).

As it can be observed from Table 3.5, utilization of neighboring motion vectors significantly improves the performance. The LK solution usually applied on the image patches which typically have well-conditioned structure tensors; therefore, conventional LK solution does not require any regularization term. However, when this solution is applied with SP region of support which could be a uniform region in general, utilization of a regularization term obtained as a weighted average of the motion vectors of neighbor SPs significantly improves the performance. Weighting the common boundary length with SP similarity or move together probability also significantly improves the performance. Move together probability is slightly better than SP similarity. Utilization of match score as an additional weight also improves the performance, but the effect is less significant. However, it should be noted that the most erroneous estimates are already eliminated in all alternatives by thresholding the match score.

Table 3.5: Effect of the regularization on LK-ICM

λ_i	$\bar{\delta}_i$	λ_w	EE
$\lambda_w \sigma_i^2 \sum_{j \in \Gamma_i} b_{ij}$	0	0.90	2.052
$\lambda_w \sigma_i^2 \sum_{j \in \Gamma_i} b_{ij}$	$\propto \sum_{j \in \Gamma_i} b_{ij}(u_j - u_i)$	0.15	1.417
$\lambda_w \sigma_i^2 \sum_{j \in \Gamma_i} s_{ij} b_{ij}$	$\propto \sum_{j \in \Gamma_i} s_{ij} b_{ij}(u_j - u_i)$	1.50	1.077
$\sigma_i^2 \sum_{j \in \Gamma_i} \lambda_s^{(ij)} b_{ij}$	$\propto \sum_{j \in \Gamma_i} \lambda_s^{(ij)} b_{ij}(u_j - u_i)$	1.20	1.074
$\sigma_i^2 \sum_{j \in \Gamma_i} \eta_j \lambda_s^{(ij)} b_{ij}$	$\propto \sum_{j \in \Gamma_i} \eta_j \lambda_s^{(ij)} b_{ij}(u_j - u_i)$	1.05	1.059

MS-ICM, LK-ICM and LK-BP are compared in terms of endpoint error and interpolation error. The parameters used in the experiments are presented in Table 3.6. Numerical results are presented in Table 3.7 and Table 3.8. Visual results are presented in Figures 3.9-3.12.

Table 3.6: Utilized parameters for the experiments

Method	λ_w	λ_s	λ_r	λ_{ovr}	λ_o	η_{th}	ϵ
MS-ICM	2.50	1	9	4	5	0.5	-log(0.5)
LK-ICM	1.05	1	9	4	5	0.5	-log(0.5)
LK-BP	-	1	9	4	5	-	-

On the first row of the Table 3.7, average power of the ground-truth motion is presented. This power corresponds to the endpoint error when the estimated motion field

is zero, which is the initial estimate for the two-frame motion estimation problem. To validate the translational motion assumption for SPs, the SP level ground-truth motion is obtained by averaging the motion vectors of the pixels in each SP. This value is the lower limit of the endpoint error for the SP based motion estimation methods with the translational motion assumption. This limit is presented on the last row of the Table 3.7. As it is shown in the table, regularized LK solution for the motion filed performs better than the MS solution, as expected, since pixel level gradients provide a rich information. LK-BP performs slightly better than LK-ICM. For three of the sequences, LK-BP results in the same solution, which indicates that LK-ICM has already converged to a local minimum in a large neighborhood where LK-BP can converge.

In Table 3.8 interpolation errors for three alternatives are presented. On the last row of the table, interpolation error obtained with the SP level ground-truth motion vectors are presented. For most of the sequences, LK-ICM and LK-BP results in a lower interpolation error, since these methods are trying to minimize the reconstruction error in a regularized manner, while SP-level ground-truth vectors are defined as to minimize the endpoint error. This results indicates that the minimum endpoint error solution is not the global minimum of the minimized energy function. Similar to endpoint error results, regularized LK solutions for the motion field performs better than the MS solution. The differences between LK-ICM and LK-BP are quite small to make a clear statement about their performance.

In Figure 3.9 and Figure 3.11 reconstruction results and interpolation errors for *Urban-2* and *Venus* sequences are presented. On the first row, the previous and current frames, on the following rows interpolation error and reconstructed current frames are presented for the three alternative approaches. Uncovered regions of the reconstructed images are highlighted with green.

In Figure 3.10 and Figure 3.12 estimated motion fields and endpoint errors for *Urban-2* and *Venus* sequences are presented. The color coding for the motion field is shown in 3.8. The endpoint errors are magnified 20 times to make them visible. The peak motion vector norm (saturation = 1 for color coding) is 22 pixels for *Urban-2* sequence and 9 pixels for *Venus* sequence. On the first row, ground-truth motion field,

and on its right SP level ground-truth motion vectors are presented. On the following rows, estimated motion vectors and corresponding endpoint errors are presented for the three alternative approaches.

Solutions employing regularized LK for the motion field solution provide better results since they are working in sub-pixel level. Especially in large and almost uniform regions LK significantly outperform MS based solutions, since MS solutions work with much less information, that is only the mean and covariance of the SPs utilized, whereas LK solution utilizes even small gradient information on these regions. As seen in the figures, BP slightly increases the accuracy of the solution.

Table 3.7: End-point error of the proposed alternatives on Middlebury test images

	Dim.	Grove2	Grove3	Hyd.	Grove	Urban2	Urban3	Venus	RMSE
AVG Power	2.17	3.13	4.54	3.91	1.35	11.65	8.51	4.20	5.89
MS-ICM	0.89	0.90	2.08	1.01	0.85	1.79	1.98	1.09	1.41
LK-ICM	**0.22**	0.52	1.76	**0.53**	0.39	1.05	**1.60**	0.42	0.98
LK-BP	**0.22**	**0.46**	**1.66**	**0.53**	0.36	**0.95**	1.65	**0.40**	**0.95**
SP Limit	0.13	0.31	0.99	0.42	0.25	0.76	0.70	0.26	0.56

Table 3.8: Interpolation error of the proposed alternatives on Middlebury test images

	Dim.	Grove2	Grove3	Hyd.	Grove	Urban2	Urban3	Venus	RMSE
MS-ICM	7.86	22.84	33.39	14.79	7.72	11.60	13.02	14.50	17.64
LK-ICM	**4.80**	14.25	24.78	**11.73**	**5.82**	7.83	**10.03**	**10.79**	12.69
LK-BP	4.82	**13.95**	**24.45**	11.82	5.88	**7.77**	10.51	10.96	**12.65**
SP GT	4.87	14.34	24.88	11.85	6.54	7.55	9.39	11.06	12.74

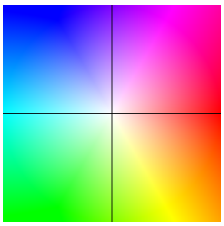

Figure 3.8: Color coding for the motion field: Direction of the motion vector is coded with the hue (red to blue), and the magnitude of the motion vector is coded with saturation (from less saturated colors to more saturated color).

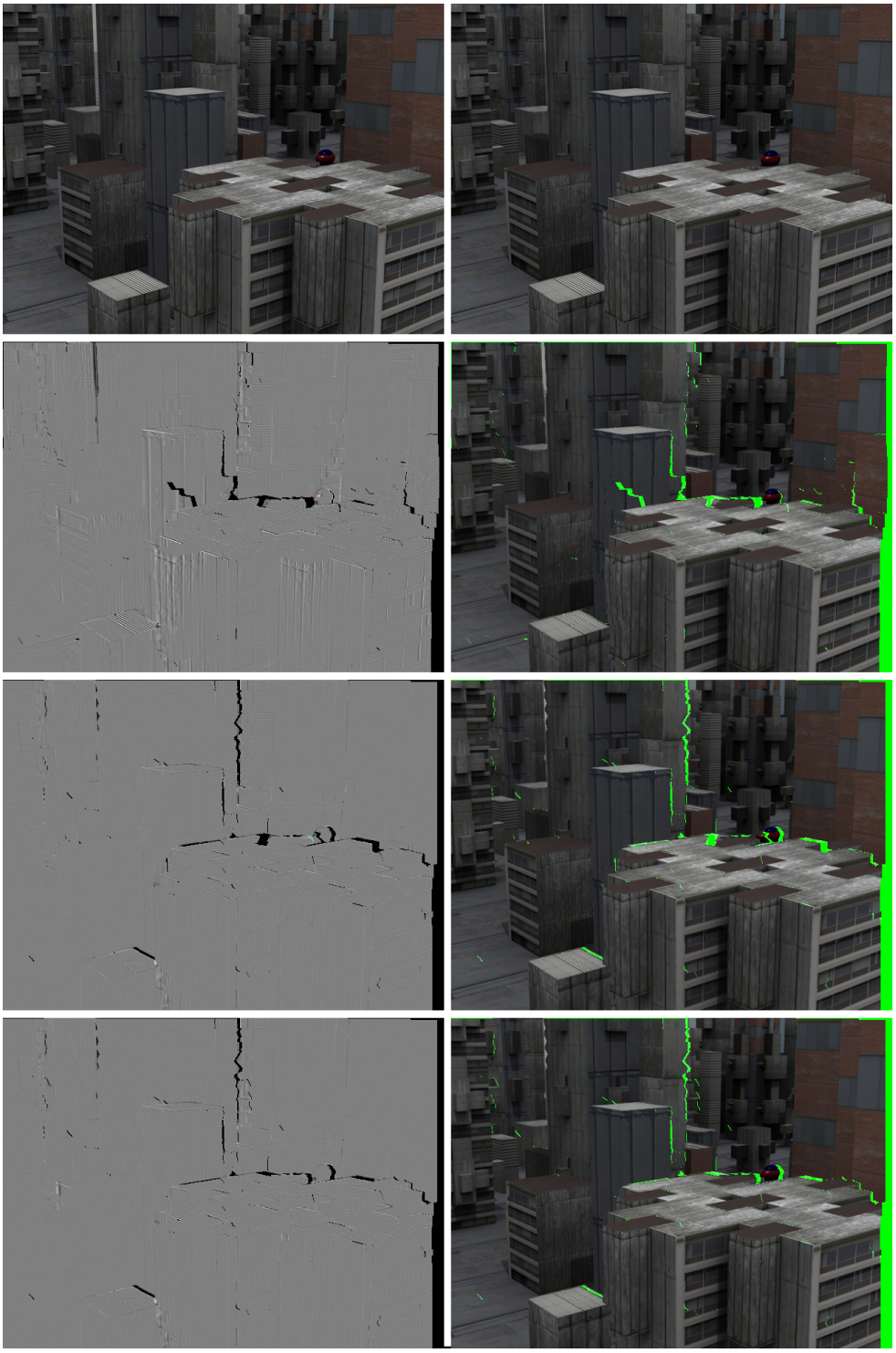

Figure 3.9: Reconstruction results on *Urban-2* sequence. On the left column: the previous image, reconstruction errors for MS-ICM, LK-ICM, and LK-BP from top to bottom, respectively; on the right column: the current frame, reconstructed current frame with the estimates of MS-ICM, LK-ICM and LK-BP from top to bottom, respectively.

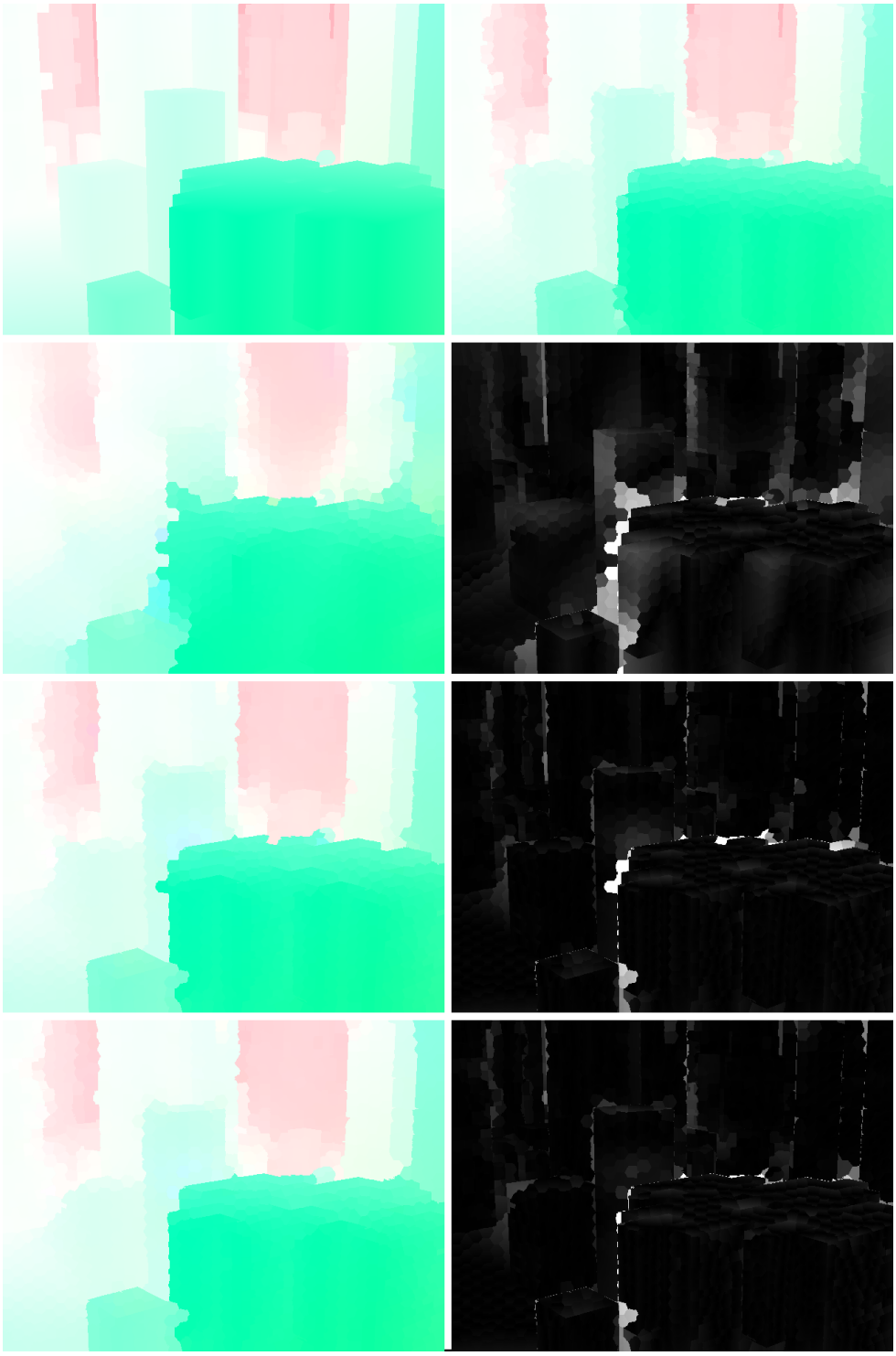

Figure 3.10: Motion estimation results on *Urban-2* sequence. On the left column: ground-truth motion vectors, results for MS-ICM, LK-ICM, and LK-BP from top to bottom, respectively; on the right column: SP level ground-truth motion vectors, endpoint error for MS-ICM, LK-ICM and LK-BP from top to bottom, respectively.

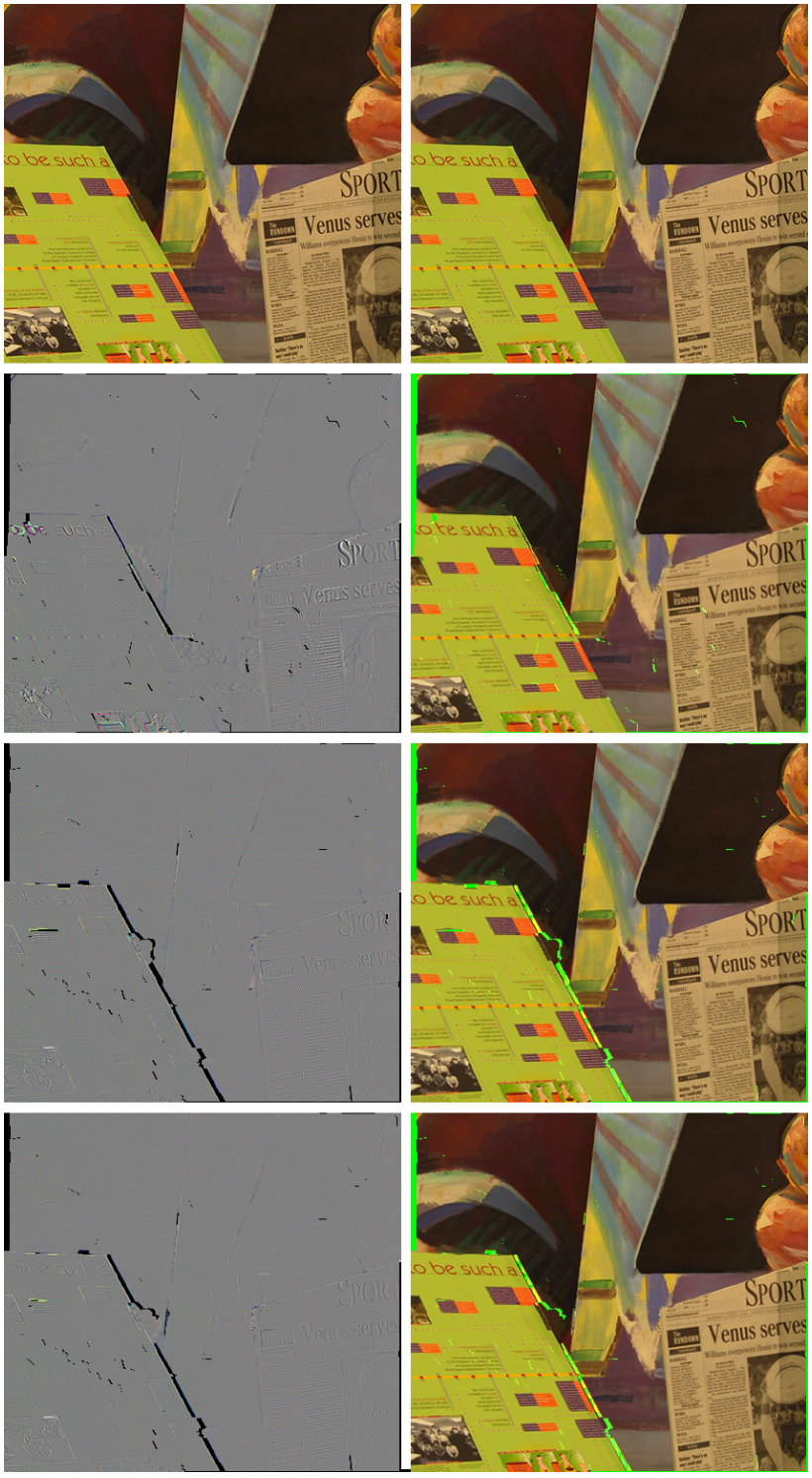

Figure 3.11: Reconstruction results on *Venus* sequence. On the left column: the previous image, reconstruction errors for MS-ICM, LK-ICM, and LK-BP from top to bottom, respectively; on the right column: the current frame, reconstructed current frame with the estimates of MS-ICM, LK-ICM and LK-BP from top to bottom, respectively.

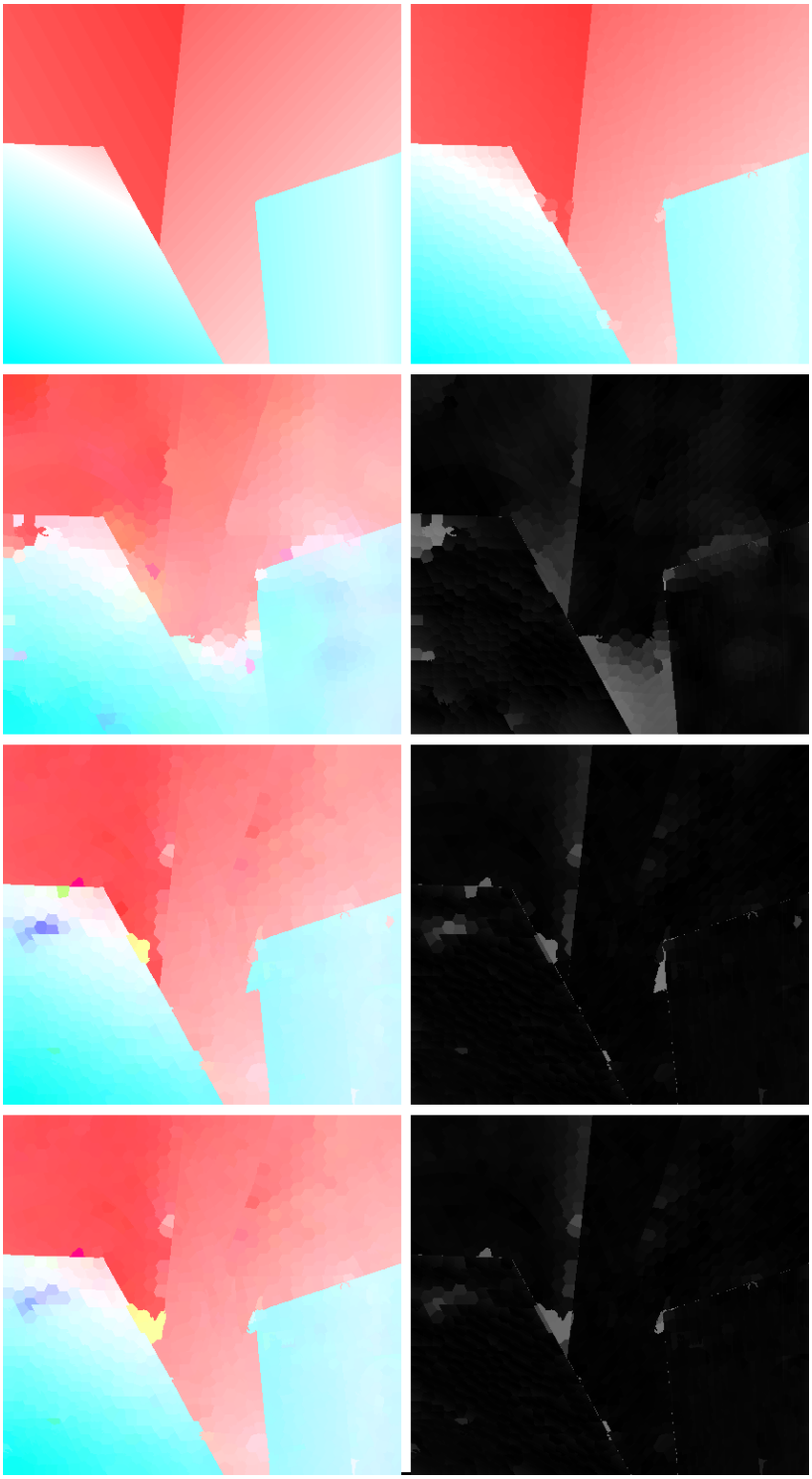

Figure 3.12: Motion estimation results on *Venus* sequence. On the left column: ground-truth motion vectors, results for MS-ICM, LK-ICM, and LK-BP from top to bottom, respectively; on the right column: SP level ground-truth motion vectors, endpoint error for MS-ICM, LK-ICM and LK-BP from top to bottom, respectively.

3.3.3 Comparative Results on Middlebury Database

LK-BP method is compared against the state-of-the-art methods in terms of end-point-error, interpolation error and overall ranking in Middlebury Database. The results are presented in Table 3.9 and Table 3.10. As shown in the tables, even if the proposed method represents whole motion field with a few hundreds of motion vectors, the performance of the proposed method is still comparable to the state-of-the-art methods.

Table 3.9: End-point error comparison on Middlebury database

	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy	Rank
NNF-Local [53]	0.07	0.15	0.18	0.10	0.41	0.23	0.10	0.34	3.4
MDP-Flow2 [54]	0.08	0.15	0.20	0.15	0.63	0.26	0.11	0.38	10.3
HAST [55]	0.07	0.18	0.17	0.15	0.49	0.58	0.19	0.32	25.0
Classic+NL [37]	0.08	0.22	0.29	0.15	0.64	0.52	0.16	0.49	34.7
SegOF [56]	0.15	0.57	0.68	0.32	1.18	1.63	0.08	0.70	81.3
LK-BP	0.19	0.52	0.62	0.47	1.11	1.57	0.15	0.97	96.6
HS [6]	0.22	0.61	1.01	0.78	1.26	1.43	0.16	1.51	105.8
Pyramid LK [6]	0.39	1.67	1.50	1.57	2.94	3.33	0.30	3.80	123.6

Table 3.10: Interpolation error comparison on Middlebury database

	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy	Rank
NNF-Local [53]	2.92	3.30	3.65	5.76	10.3	6.42	7.57	7.61	27.8
MDP-Flow2 [54]	2.89	3.47	3.66	5.2	10.2	6.13	7.36	7.75	12.6
HAST [55]	3.01	3.45	6.39	5.43	11.2	7.47	8.68	8.35	56.2
Classic+NL [37]	3.1	3.66	4.78	5.36	11.5	6.73	8.74	8.29	62.2
SegOF [56]	3.51	4.17	8.69	8.58	11.7	6.79	10.1	8.8	100.5
LK-BP	3.73	4.37	6.07	6.26	11.3	6.89	10.55	8.22	98.7
HS [6]	3.16	4.91	6.13	6.80	10.9	6.16	8.63	7.91	65.2
Pyramid LK [6]	4.16	5.83	11.4	12.4	14.3	6.69	10.3	11.1	114.2

In order to make a more fair comparison, results of some sample algorithms in Table 3.9 are quantized at SP-level, that is for each SP the motion vector is obtained by averaging the motion vectors of the pixels in the region of support of the corresponding SP. Since the ground-truth for the evaluation set is not publicly available, comparison

is performed on test images. The endpoint errors of the original algorithms and the endpoint errors for SP-level quantized motion field are presented in Table 3.11. As seen in the table, SP-level quantization of the motion field generally increases the endpoint error; however, quantization might also reduce the endpoint error, probably by low pass filtering like effect for erroneous estimates. These results also validate the translational assumption for SP motion. Even the increase in endpoint error is small due to the SP-level quantization, since most of the algorithms are resulting in quite close estimates, such small deviations affect the ranking of the algorithms.

Table 3.11: SP-level endpoint errors of some methods on Middlebury test images

	Dim.	Grove2	Grove3	Hyd.	Grove	Urban2	Urban3	Venus	RMSE
MDP-Flow2 [54]	0.24	0.32	1.32	0.38	0.26	0.69	1.61	0.34	0.81
MDP-Flow2 SP	0.25	0.36	1.39	0.45	0.30	0.84	1.61	0.37	0.86
Classic+NL [37]	0.16	0.29	1.29	0.40	0.26	0.81	1.26	0.43	0.74
Classic+NL SP	0.19	0.35	1.38	0.46	0.29	0.88	1.33	0.44	0.80
SegOF [56]	0.23	0.43	1.56	0.34	0.31	1.59	2.09	0.93	1.23
SegOF SP	0.25	0.40	1.56	0.45	0.35	1.56	2.08	0.89	1.14
LK-BP	0.22	0.46	1.66	0.53	0.36	0.95	1.65	0.40	0.95

3.4 Conclusions

In this chapter, a SP-based occlusion aware layered motion estimation method is proposed. The proposed method solves the local layer orders of SPs and the sparse motion field which is the union of the SP displacements. The layer order solution substitute the conventional line field in the dense motion estimation problem, and when combined with the motion vectors, they provide the pixel level occlusion field.

The motion estimation problem is expressed as a MAP estimation problem, and a Bayesian formulation for the MAP estimation is presented. For the layer order solution, an ICM-based method is proposed, and for the motion field solution, well-known Mean-shift and optical flow methods are extended for SP-based motion estimation. For the Mean-shift method, the kernel is defined with the SP region of support and the spectral mean and variance of the SP on the previous frame. Image gradient based optical flow solution is performed with the arbitrary region of support obtained from

SPs. For both methods, the region of support for SPs is updated throughout the iterations using the layer order solution to handle the occlusions. Particularly for the ICM-based solution, expected value of the neighboring motion vectors are calculated more precisely by using adaptive weighting between neighbors.

For the joint optimization of the layer orders and the motion field, a global optimization approach is proposed based on particle belief propagation. For this solution, rather than generating the particles by random sampling around the possible states, results of the ICM-based solution, which can converge to local minimum in a large neighborhood, is utilized. Performing the belief propagation on the network constructed by the SPs, rather than the individual pixels, greatly reduces the computational complexity for the global optimization. Utilization of ICM-based solution for the birth process makes it possible to work with a few particles for each SP, which reduces the computational complexity further.

For the motion field solution, neighborhood relations and quality if the motion estimates are exploited extensively, to achieve an effective regularization. At first, the the ambiguity due to the under-determined set of equations in optical flow constraint is reduced by utilizing the SPs as the region of support. Furthermore, utilizing the neighboring motion vectors in an effective and adaptive way to regularize the solution, quite accurate results are obtained.

Particle belief propagation solution slightly increases the performance. Improvement in the performance is due to the joint optimization; however, the improvement is not significant. This may be due to the either the utilization of the motion vectors generated by the ICM-based solution, forcing BP solution to converge to the same local minimum. As the results of the ICM-based solution are already quite close to the true solution, it would be hard for BP to find significantly better results.

The proposed method differs from previous work in terms of the following points:

- Solutions in [39, 42] need a high-quality dense motion estimate to solve the problem, while proposed method does not require any motion estimate.
- In proposed method, the factorization for the conditional density is different from [42], which yields to a more efficient solution.

- The proposed method utilizes pixel level gradients rather than SP level gradients in [41] to improve the accuracy.
- Utilization of layer orders like in [42] helps handle the occlusions and the additional moving away option preserve motion discontinuities better, while other gradient descent approaches like [41] suffer from occlusions and motion discontinuity boundaries, and [39] is not able to provide explicit optical flow solution due to lack of an occlusion model.
- One-to-one matching of SPs of two different frames is not possible for the methods that apply SP extraction independently in two frames, such as [40] which works with multiple hypotheses to solve this ambiguity, while proposed method can be utilized to obtain one-to-one matching for SPs and preserves point-to-point matching between the frames.
- In the proposed method, gradient descent methods are utilized for updating the motion field for each particle and belief propagation is performed with these particles, while [42] samples around the most probable states and performs belief propagation with these particles iteratively.

Performance of the proposed method is measured on Middlebury database, and the proposed method shown to generate quite accurate motion estimates which are almost comparable to the state-of-the-art dense motion estimation algorithms. The proposed method also has a low computational complexity, which is slightly higher than the classical dense method proposed by Horn and Schucnk [25].

One of the most important advantages of the proposed method is to express the dense motion field with a few number of motion vectors. The proposed method can be utilized for temporally consistent SP extraction, and various applications like compression, video object segmentation or object tracking.

CHAPTER 4

TEMPORALLY CONSISTENT SUPERPIXELS

Superpixels (SPs) provide an efficient representation for the still images. The recent SP extraction methods [16, 18] and the proposed SP extraction method in Chapter 2 are quite fast and accurate. However, they do not provide consistent segmentation results between frames, which are essential for motion estimation and useful for segmentation, unless the association of SPs is performed by another algorithm. The volumetric extensions of these algorithms may result in SPs having similar shapes throughout the video; however, they are not able to preserve the point-to-point matching between frames which is one of the main requirements for temporally consistent SPs.

Performing SP extraction for each frame independently, grouping them in object level with-SP based segmentation methods [16] and estimating the motion by association of these objects might be applicable for object level segmentation, but will require complex motion models and such an approach would not be able to substitute temporally consistent SPs for some certain applications, such as video compression.

Joint segmentation and motion estimation methods [8, 39] result in temporally consistent SPs; however, they are computationally complex and not applicable to online video processing. SP level motion estimation methods, such as [41] and the one proposed in Chapter 3, can be utilized for propagating the SPs obtained in the previous frames throughout the video for obtaining temporally consistent SPs.

In this chapter, a method for consistent SP extraction is proposed which utilizes SP level motion estimation results. In the first section, related work for consistent SP ex-

traction is presented. Proposed consistent SP extraction method and its performance are explained in the following sections.

4.1 Related Work

Considering the consecutive frames as a volume and applying supervoxel (SV) extraction methods [16, 18] might be a solution for obtaining SPs which preserve their shapes along an image sequence; however, SV methods do not preserve point-to-point matching, are usually complex and are not applicable to online video processing. Moreover, independently moving clusters or large displacements (i.e. objects whose SPs do not overlap on consecutive frames) can not be handled by these methods.

In a preliminary work [39], the definition of temporal SP is given and a joint method for extracting temporal SPs is presented. Another joint method [8], computing the optical flow and performing segmentation jointly, is proposed and utilized for obtaining consistent segmentation results.

SP-based motion estimation methods, solving the motion field from the previous frame to the current frame by using the SPs on the previous frame, inherently provide temporally consistent SPs, except remaining some regions uncovered in the current frame. Moreover, any dense motion estimation algorithm can be utilized to obtain SP level displacements and consistent SP extraction problem is reduced to solve local layer ordering to handle the occlusions [42] and labeling of uncovered regions. SP extraction and motion estimation methods are reviewed in Chapter 2 and Chapter 3, respectively. For the joint solutions of temporally consistent SP extraction and SP based motion estimation methods, readers can refer to Section 3.1.3.

If a dense solution is utilized for the motion estimation step, then SP extraction can be performed with the method proposed in Chapter 2 or other SP extraction methods, such as [16, 18]. These methods can utilize the dense motion information by appending the motion vectors to the feature vector which contains the spectral and spatial information:

$$X = [I_1(x,y) \ \dots \ I_c(x,y) \ x \ y \ u_x(x,y) \ u_y(x,y)]^T \tag{4.1}$$

where $[I_1(x,y) \ \ldots \ I_c(x,y)]$ is observed visual data over c image channels (i.e. RGB, Lab, etc.) and $[x \ y]$ is the position on the image plane, and $u_x(x,y)$ and $u_y(x,y)$ are the elements of the motion vector for the given position. The SP extraction method is not in the scope of this chapter, for the rest of the chapter SPs on the first frame is assumed to be available.

4.2 Consistent Superpixel Extraction

Temporally consistent SPs are small connected regions, preserving the local structure on each image, and point-to-point matching and SP shapes throughout the image sequence. For temporally consistent SP extraction, if there are no occlusions and uncovered regions, then the number of SPs should be constant. When there are occlusions and uncovered regions, the number of SPs might change. In both cases, SP shapes might evolve to preserve the local structure; however, point-to-point matching between frames should also be preserved. To achieve such a result, SPs on the first frame might be propagated and their shapes should be evolved throughout the image sequence.

In Chapter 3, it is shown that the proposed SP based motion estimation method provides quite accurate results. The displacements of the SPs can be obtained with such an approach, which provides the set of SP displacements $\left(\{\boldsymbol{u}_i\}_{i=1}^{N}\right)$ from the previous frame, I_{k-1}, to the current frame, I_k.

When the translational model assumption is invalid (e.g. different manifolds), an appropriate motion model for the imaging device, such as planar or affine, should be utilized. For the rest of the Section, SP motion will be assumed to be translational and available, which is an acceptable assumption for small-sized SPs.

For the translational SP displacement assumption, if a dense motion field, $\boldsymbol{U}$, from the previous frame to the current frame and SPs on the previous frame are given, SP motion can be calculated as (4.2):

$$\boldsymbol{u}_i = \frac{1}{|\Omega_{i,k-1}|} \sum_{x \in \Omega_{i,k-1}} U(\boldsymbol{x}) \tag{4.2}$$

where $\boldsymbol{x}$ is the spatial position vector and $\Omega_{i,k-1}$ is the region of support for SP i on

the previous frame.

Given the SP displacements, the label image in the current frame can be initiated with the SPs in the previous frame with three steps: pixel to cluster assignment, generation of new clusters and assignment of unlabeled pixels. Once the SPs on the current frame are initiated, they can be refined by using any SP extraction algorithm, such as the proposed LASP or SLIC [18].

Pixel to Cluster Assignment

Given the SP displacements, the pixels occupied by SP i on the current frame can be obtained by the integer casted motion vectors, as in Chapter 3:

$$\hat{\Omega}_{i,k} = \{x | x - \tilde{u}_i \in \Omega_{i,k-1}\} \tag{4.3}$$

where $\tilde{u}_i$ is the integer casted displacement from the previous frame to the current frame forSP i. Once the SPs are placed on the current frame, overlapping regions can be obtained by the following relation:

$$\Omega_{ij} = \hat{\Omega}_{i,k} \cap \hat{\Omega}_{j,k} \tag{4.4}$$

The pixels in the overlapping region should be assigned according to the local layer order. If two SPs are overlapping, then one of them should be occluding the other. Occluding SPs is selected by minimizing the reconstruction error in the overlapping region. The reconstruction error in overlapping region is obtained by the difference between the current frame and the interpolated previous frame:

$$J_r(i) = \sum_{x \in \Omega_{ij}} \|I_k(x) - I_{k-1}(x - u_i)\|^2 \tag{4.5}$$

The local layer order for the overlapping pixels is selected by minimizing the reconstruction error:

$$R_{ij} = \begin{cases} 1 & if\ J_r(i) < J_r(j) \\ -1 & o.w. \end{cases} \tag{4.6}$$

The initial region of support of SPs on the current frame is obtained by excluding the occluded regions:

$$\Omega^{(0)}_{i,k} = \{x | x \in \hat{\Omega}_{i,k}\ and\ R_{ij} = 1\ \forall\ j\ st. x \in \hat{\Omega}_{j,k}\} \tag{4.7}$$

Using the initial region of support, the initial label image can be obtained:

$$\hat{L}(x) = \begin{cases} i & if\ \exists\ i\ st.\ x \in \Omega_{i,k}^{(0)} \\ 0 & o.w. \end{cases} \tag{4.8}$$

Generation of New Clusters

When there is a camera movement in the scene, boundary regions of the previous image should move out of the image, and the regions which are not observed in the previous image should enter to the current frame. In order to preserve the number of SPs and to have the SPs with similar sizes, new clusters should be generated. An example of adding new clusters is shown in Figure 4.1. In this sequence, the camera is panning to the left, so whole scene is moving to the right. The left boundary of the current image is not present in the previous frame. Moving the SPs with their estimated displacements and checking the possible neighbors (which are out of the image in previous frame) of the boundary SPs, whether they are in the current image or not, new SP decision is given. Centers of new SPs are shown with red marks.

Assignment of Unlabeled Pixels

Since SPs might move independently due the to given motion estimates, estimated label image in (4.8) might have unlabeled pixels (holes), as shown in Figure 4.1. Those unlabeled pixels are assigned to the nearest SP with respect to the spatial Mahalanobis distance:

$$L(x) = \begin{cases} \arg\min_i (x - \bar{x}_i)^T \Sigma_i (x - \bar{x}_i) & if\ \hat{L}(x) = 0 \\ \hat{L}(x) & o.w. \end{cases} \tag{4.9}$$

where $\bar{x}_i$ and Σ_i are the spatial mean and covariance for SP i. Once the initial label assignment is completed, this initial estimate is refined by the proposed LASP algorithm. A typical example of initial label assignment and the output of LASP algorithm are presented in Figure 4.2.

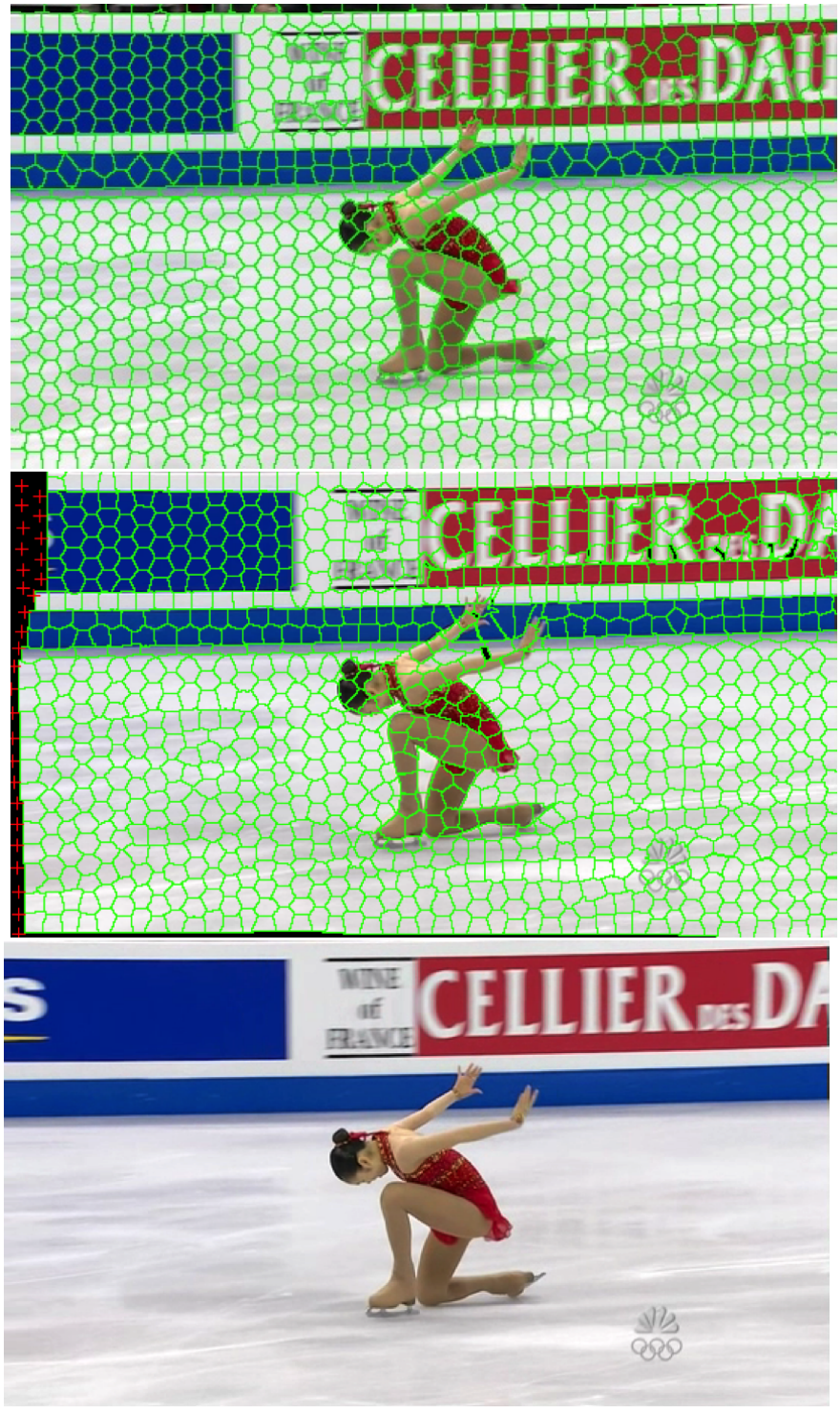

Figure 4.1: Initiation of the current frame label image with the previous frame SPs and the estimated displacements. From top to bottom: the previous frame, reconstructed current frame with SPs in the previous frame on which the new SP centers marked by red cross, an the current frame.

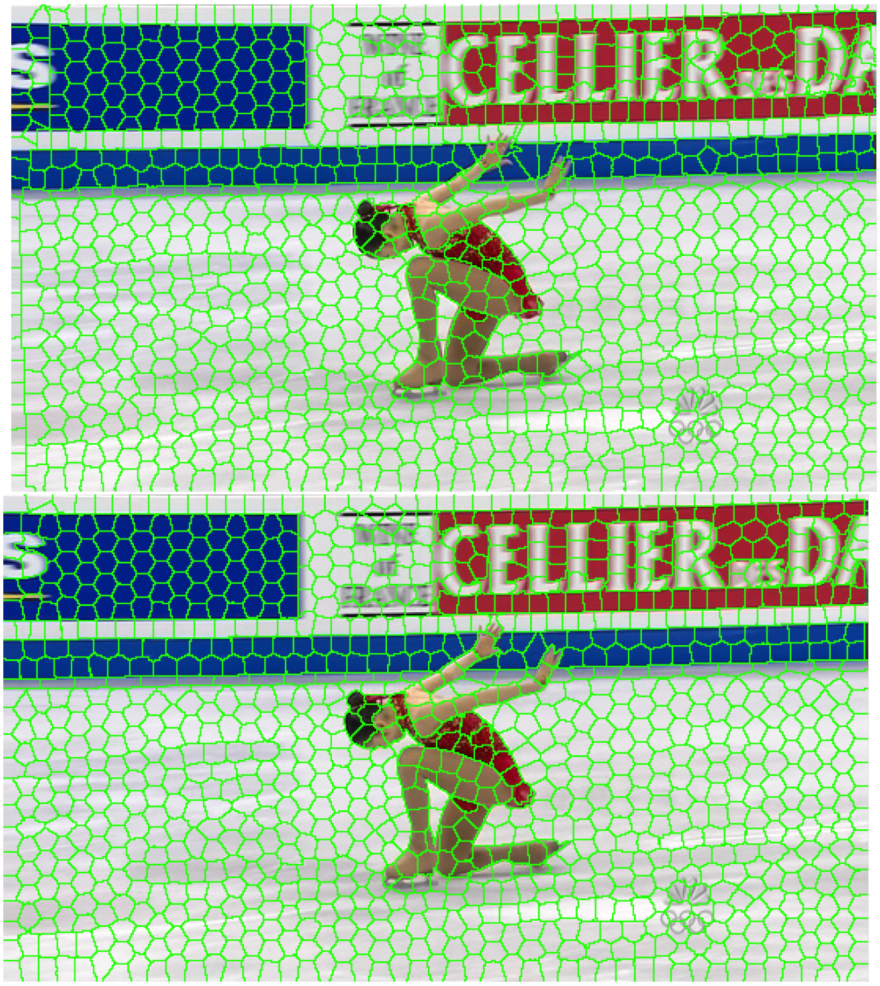

Figure 4.2: Initiation of the current frame SPs (top) and current frame SPs after LASP iterations (bottom).

4.3 Experiments

The effectiveness of the proposed consistent SP extraction method can be measured by comparing the performance of temporally consistent SPs with the SPs obtained on each frame independently. Algorithms are tested on eight sequences from [57]: *bowling, campanile, deoksugung, drone, galapagos, hippo fight*, and *horse riding*, with 1000 SPs, hexagonal honey comb tiling and adaptive weights in LASP algorithm.

Boundary recall, bleeding error and number of pixel update controls for independently obtained SPs and temporally consistent SPs are presented in Figures 4.3 - 4.5. Temporally consistent SPs perform slightly better both for boundary recall and bleed-

ing error. The proposed LASP algorithm checks for pixel label update on the SP boundaries, and excludes boundaries of the settled SPs. As shown in 4.5, due to the high quality initial estimates, the temporally consistent SPs converge more quickly; hence, LASP excludes most of the boundary pixels for pixel label update iterations. The results are summarized in Table 4.1.

Table 4.1: Results for independent and temporally consistent SP extraction.

	Independent SPs	Temporal SPs
Boundary recall	0.945	**0.936**
Bleeding error	0.188	**0.168**
# Pixel operations	1252429	**320137**

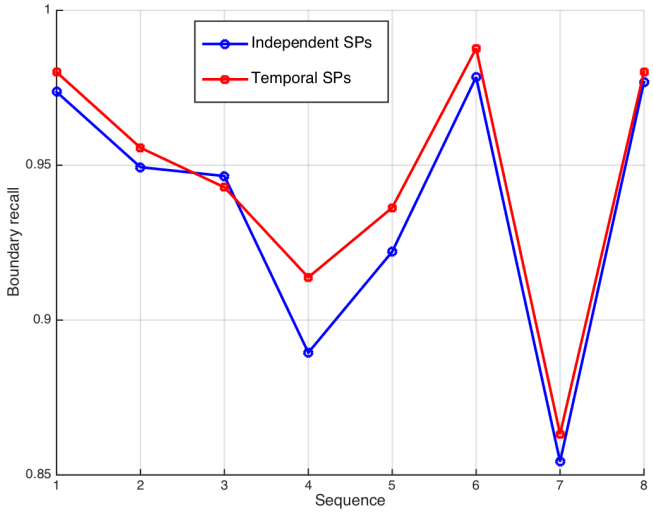

Figure 4.3: Boundary recall for independent SPs and temporal SPs.

Sample frames from the *drone* and *horse riding* sequences are presented in Figures 4.6 and 4.7, where the independent SPs are on the left and the temporally consistent SPs are the right column, respectively. As shown in the figure, the temporal SPs preserve their shapes as well as consistently covering the similar regions between the frames, whereas the regions covered by independent SPs and their shapes are suddenly changing between the frames.

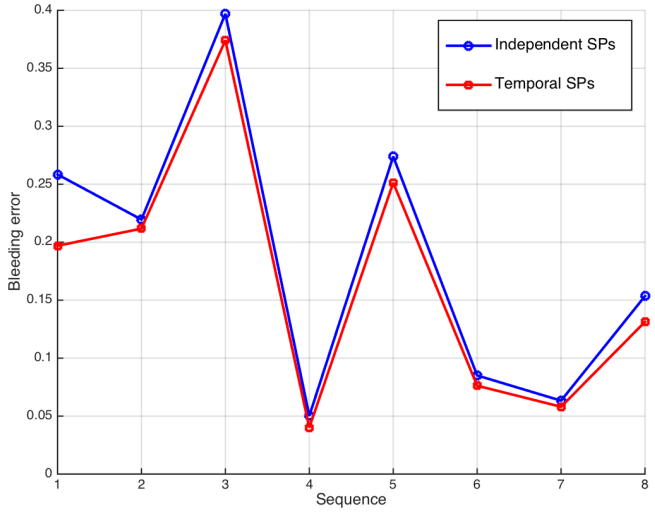

Figure 4.4: Bleeding error for independent SPs and temporal SPs

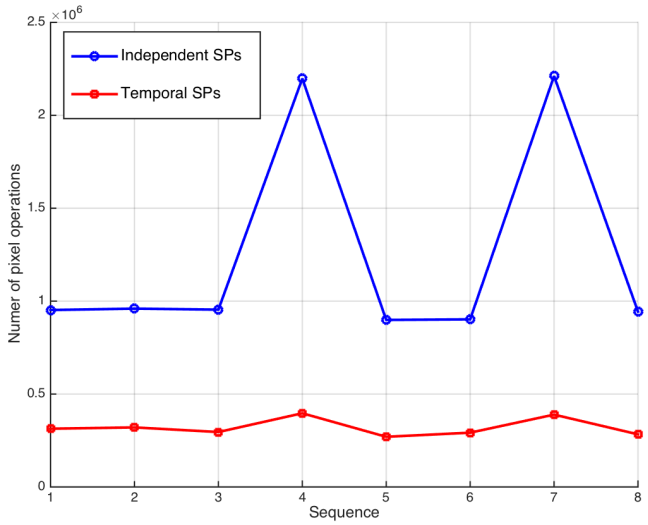

Figure 4.5: Number of pixel update controls for independent and temporal SPs.

4.4 Conclusions

In this chapter, a general formulation for temporally consistent SP extraction with a propagating solution is presented. The proposed method is based on the utilization of SP level motion estimates and initiation of the SPs on the current image with the SPs in the previous image. The proposed method can utilize any SP extraction method for obtaining the SPs on the previous frame and refining the initial estimates in the current frame. For the motion estimation, any dense motion estimation method or SP

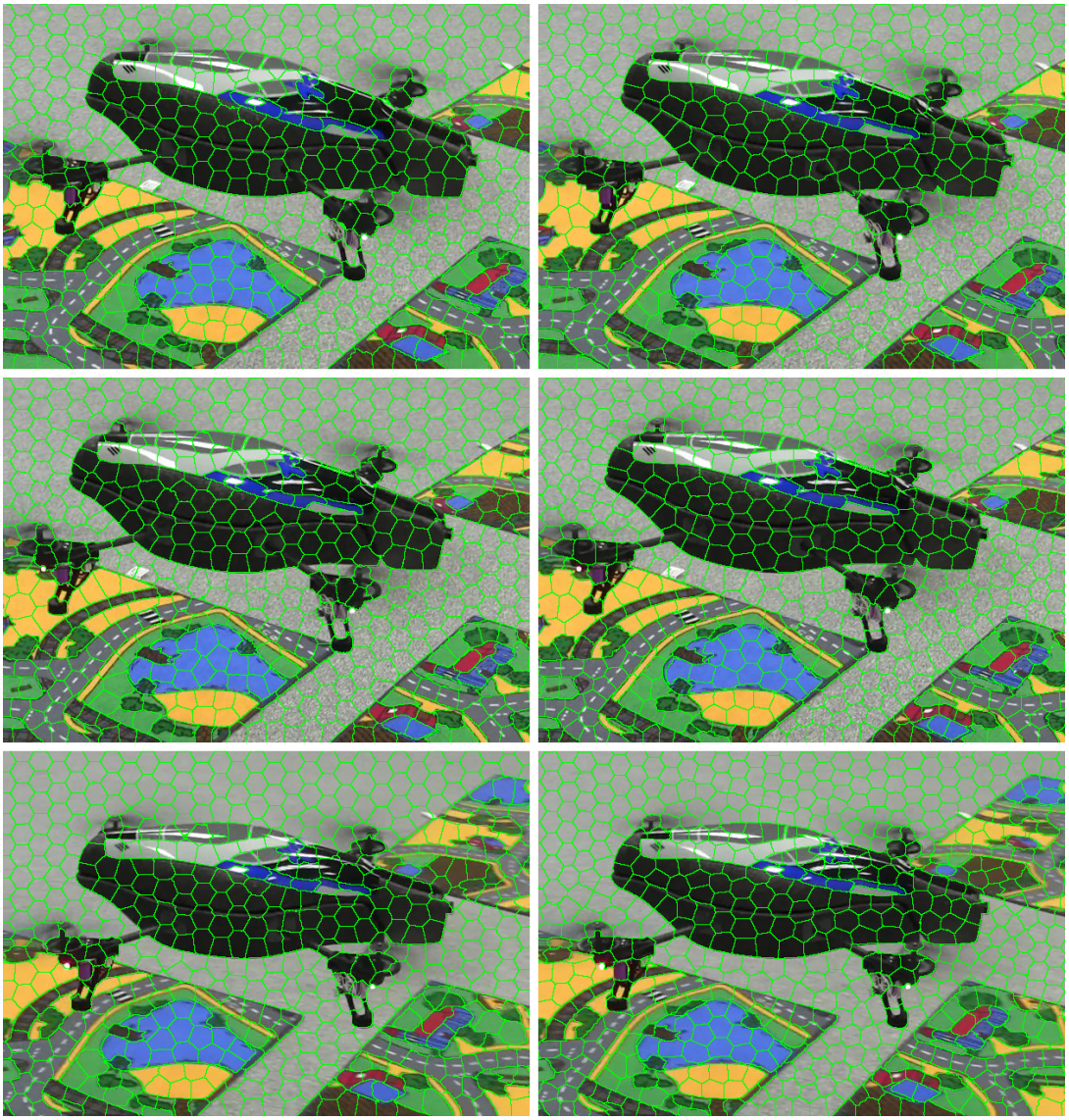

Figure 4.6: Temporally consistent SPs on *drone* sequence: independent SPs on the left column, temporally consistent SPs on the right column.

level motion estimates can be utilized. For SP extraction, the proposed SP extraction method in Chapter 2 and for the motion estimation the proposed SP level motion estimation method in Chapter 3 are utilized.

The performance of the proposed temporally consistent SP extraction method is compared against the independent solution, in terms of boundary recall, bleeding error and pixel label update controls. The accuracy of the proposed method is slightly better than the independent SP extraction approach, due to the high-quality initial

Figure 4.7: Temporally consistent SPs on *horse riding* sequence: independent SPs on the left column, temporally consistent SPs on the right column.

estimate for the label image. Having a high-quality initial estimate also helps algorithm to converge more quickly, reducing the number pixel label update controls; hence, speeds-up the SP extraction step. There is an additional cost of the motion estimation algorithm; however, for the applications which require the matching of SPs throughout an image sequence, motion estimation step can be afforded instead of an association. The proposed solution results in both a more accurate segmenta-

tion and inherently matched SP. Temporally consistent SPs can be utilized for various computer vision applications such as segmentation, compression, object tracking and background modeling.

CHAPTER 5

CONCLUSION

In this thesis, building blocks for superpixel-based image sequence representation is developed. Superpixels (SPs) are exploited extensively, for still image, image pair and image sequence representation and motion estimation.

For the still images, a novel gradient ascent method for SP extraction is proposed. The proposed SP extraction method, utilizes first and second order spectral and spatial statistics of SPs to achieve an optimal Bayesian classifier for pixel to SP label assignment, which reduces the dependency on user selected parameters. With the refinement of the initial grid and the optimizations in the gradient ascent iterations, a relatively fast algorithm is achieved. The computation time of the proposed method is hardly affected by the number of SPs, making it easier to utilize the larger number of SPs. With the utilization of honey-comb hexagonal initial tiling instead of a regular square grid, algorithm results in more convex shaped SPs and less number of neighboring SPs. Having less number of neighbors would simplify the graphs constructed by SPs, hence reduce the computational complexity. Achieving more convex shaped SPs made it easier to utilize SPs for motion estimation. The proposed method, Local Adaptive Superpixels (LASP), is compared against the state-of-the-art methods in Berkeley segmentation database. LASP is shown to outperform other gradient ascend methods in terms of boundary recall, bleeding error and computation time.

An SP-based layered occlusion aware motion estimation method is proposed to propagate the SP solution throughout the image sequences and to obtain temporally consistent SPs with a computationally efficient way. The motion estimation problem is defined as a MAP estimation problem, in which the conventional line and occlusion

fields are replaced with the local layer orders by the utilization of SPs. The proposed motion estimation method solves the local layer orders and the motion field. Layer orders helps to handle occlusions and the proposed novel moving away layer order helps to preserve motion discontinuity boundaries. The ambiguity in optical flow constraint equation is reduced by utilizing SPs as the region of support for motion vectors. Moreover, neighbor SPs are utilized in an effective and adaptive way to regularize the motion field solution. However, using the SPs as the region of support, necessarily brings a parametric motion assumption, hence estimates should be more precise to achieve a similar accuracy with the state-of-the-art methods. Modeling the occlusions and selecting the smoothness weights for regularization by exploiting the SP neighbor relations such as common boundary length and SP similarity, more precise results are achieved. Utilization of SPs as the region of support also made it possible to evaluate the quality of the motion estimates, which help to correct the initial estimates of motion vectors for large displacements and small-sized SPs. With the correction of these initial estimates, it become possible to converge to an accurate solution. Quality of the motion estimates are also utilized for the regularization, inherently resulting an adaptive and anisotropic smoothness weight.

For the solution of the local layer orders, Iterated Conditional Modes is employed, and for the occlusion case ambiguity caused by erroneous motion estimates is utilized to temporarily allow an illegal local layer ordering and correct the motion estimates via regularization. For the given local layer orders, mean-shift and regularized LK approaches are adapted for SP level motion estimation. Mean-shift approach results in coarser estimates, but handles larger displacements, while regularized LK approach has a higher precision. The ICM-based solution is a fast solution alternative for the local layer orders.

For a given a set of motion vectors for each SP, a global optimization scheme based on particle belief propagation is also proposed for the joint optimization of the local layer orders and the motion field. In order to obtain a computationally efficient algorithm, ICM-based solution is utilized for the birth process of the particles in particle belief propagation. Particle belief propagation method results in a slightly better motion field.

As the independent extraction of SPs through an image sequence results in a temporal representation deficiency, in order to achieve a temporally consistent representation, a general approach based on propagating the SP information throughout an image sequence is also proposed. The displacement of SPs between frames are assumed to be translational and SPs are assumed to be rigid, which enables to move SPs from one frame to the next one. The proposed motion estimation method is utilized to obtain the SP displacement. Using the SP displacements, SPs on the previous frame are placed on the current frame, occlusions are determined by minimizing reconstruction errors to initiate the label image. For the uncovered regions either new SPs are generated or pixels in these regions are assigned to nearest SPs to have the initial SPs which cover the whole image. The proposed SP extraction method is utilized for obtaining the SPs in the first frame and correcting the initial label image in the following frames. The proposed temporally consistent SPs keeps the point-to-point matching, evolves to preserve local structure while trying to preserving their shapes throughout the image sequence as much as possible.

5.1 Future Work

The proposed SP extraction algorithm LASP is shown to outperform state-of-the-art gradient ascent methods; however, the current performance measures do not distinguish the performances of the methods in terms of the convexity and regularity of SPs. Moreover, the current metrics, bleeding error and boundary recall, do not help to make certain conclusions on the SP based segmentation performance. In order to obtain the bleeding error (also called as the under-segmentation error) an SP is assigned to the foreground object if the SP intersects with the object. This is the measure of the precision when the recall rate is set to one. If it is aimed to conclude about the segmentation performance, then the maximum achievable segmentation accuracy should be considered first. For this measure, each SP should be assigned to the object (foreground or background) with the maximum intersection area. Other alternative metrics might also be studied to make more general conclusions about the segmentation performance.

Another improvement might be utilization of simple pixel control as proposed in [58].

This approach eliminates the necessity for connected component labeling (CCL) at the end of the pixel to SP label assignment iterations. Such a control reduces the computational complexity due to CCL; however, there will be an overhead due to this control. More importantly, enabling simple-pixel control might further reduce the dependency on user selected parameters.

The proposed motion estimation method might be tested with different SP extraction algorithms to evaluate its robustness against the errors in SP extraction. The method should also be evaluated on different databases, especially on those which include large displacements. The performance of the proposed motion estimation method should be investigated further in terms of computational complexity.

The proposed motion estimation method is able to handle occlusions and large displacements. However, if the displacement is large such that the initial estimate does not overlap with the SP on the current frame, then mean-shift approach cannot converge and if SP is independently moving, then the displacement of SP cannot be initiated with its neighbors. For those SPs either a search algorithm should be employed or data association methods should be utilized.

For the motion estimation method, rather than the brightness consistency assumption, brightness and contrast changes in the scene might be solved as the global offset and gain parameter. In this case, there will be three main steps in the algorithm: the solution of layer orders, the solution of the offset and gain, and the solution of the motion field. Such an extension would result in a more general motion estimation algorithm.

For temporally consistent SP extraction, no performance metric is proposed yet. In order to evaluate the performance of different methods, a performance metric should be defined.

For long-term temporally consistent SP extraction, tracking and data association methods might be utilized. Modeling the observed motion as the sum of the global camera movement and individual SP movements with n^{th} order constant derivatives, Kalman filters or Kalman smoothers might be employed. The appearance changes in SPs might also be modeled and tracked for temporally consistent SP extraction. If such

methods are utilized for tracking the SP trajectories or appearances, then a SP mapping between frames should also be evaluated. Whenever the motion estimation algorithm provides high-quality motion estimates, the mapping is one-to-one. However, for the mismatched SPs, large displacements for which the motion estimation algorithm fails, disappearing SPs and new born SPs an SP matching should be performed.

REFERENCES

[1] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[2] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, September 2004.

[3] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[4] Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(6):583–598, 1991.

[5] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[6] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[7] Steven M Seitz and Simon Baker. Filter flow. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 143–150. IEEE, 2009.

[8] C.L. Zitnick, Nebojsa Jojic, and Sing Bing Kang. Consistent segmentation for optical flow estimation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1308–1315 Vol. 2, Oct 2005.

[9] Samuel S Blackman. Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1, 1986.

[10] Yaakov Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., 1987.

[11] Bar-Shalom Yaakov. *Multitarget-multisensor tracking: Applications and advances*. Boston; London: Artech House, 1992.

[12] Samuel Blackman and Robert Popoli. Design and analysis of modern tracking systems. *Norwood, MA: Artech House*, 1999.

[13] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 10–, Washington, DC, USA, 2003. IEEE Computer Society.

[14] H. Emrah Tasli, Cevahir Cigla, and A. Aydin Alatan. Convexity constrained efficient superpixel and supervoxel extraction. *Signal Processing: Image Communication*, 33:71 – 85, 2015.

[15] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2290–2297, December 2009.

[16] Cevahir Çığla and A Aydın Alatan. Efficient graph-based image segmentation via speeded-up turbo pixels. In *2010 IEEE International Conference on Image Processing*, pages 3013–3016. IEEE, 2010.

[17] Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Vincent Lepetit, and Pascal Fua. A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images. In *Proc. Medical Image Computing and Computer Assisted Intervention (MICCAI 2010), Part II*, volume LNCS 6362 of *Lecture Notes in Computer Science*. Springer, 2010.

[18] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, November 2012.

[19] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 13–26, Berlin, Heidelberg, 2012. Springer-Verlag.

[20] Thomas C. Hales. The honeycomb conjecture. *Discrete and Computational Geometry*, 25(1):1–22, 2001.

[21] Jie Wang, Caiming Zhang, Yuanfeng Zhou, Yu Wei, and Yi Liu. Global contrast of superpixels based salient region detection. In *Computational Visual Media*, pages 130–137. Springer, 2012.

[22] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[23] Gang Zeng, Peng Wang, J. Wang, Rui Gan, and Hongbin Zha. Structure-sensitive superpixels via geodesic distance. In *2011 International Conference on Computer Vision*, pages 447–454, Nov 2011.

[24] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *in Proc. 8th Int'l Conf. Computer Vision*, pages 416–423, 2001.

[25] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[26] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[27] Janusz Konrad, Eric Dubois, et al. Bayesian estimation of motion vector fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.

[28] Manuel Werlberger, Werner Trobin, Thomas Pock, Andreas Wedel, Daniel Cremers, and Horst Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, volume 1, page 3, 2009.

[29] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer, 2009.

[30] Andreas Wedel, Daniel Cremers, Thomas Pock, and Horst Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1663–1668. IEEE, 2009.

[31] Henning Zimmer, Andrés Bruhn, Joachim Weickert, Levi Valgaerts, Agustín Salgado, Bodo Rosenhahn, and Hans-Peter Seidel. Complementary optic flow. In *Energy minimization methods in computer vision and pattern recognition*, pages 207–220. Springer, 2009.

[32] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[33] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[34] E Dubois and J Konrad. Estimation of 2-d motion fields from image sequences with application to motion-compensated processing. In *Motion analysis and image sequence processing*, pages 53–87. Springer, 1993.

[35] A. Murat Tekalp. *Digital Video Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[36] Berthold K.P. Horn and B.G. Schunck. "determining optical flow": a retrospective. *Artificial Intelligence*, 59(1–2):81 – 87, 1993.

[37] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[38] Gerard De Haan, Paul WAC Biezen, Henk Huijgen, and Olukayodc A Ojo. True-motion estimation with 3-d recursive search block matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(5):368–379, 1993.

[39] Jason Chang, Donglai Wei, and John W Fisher. A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058, 2013.

[40] Amelio Vazquez-Reina, Shai Avidan, Hanspeter Pfister, and Eric Miller. Multiple hypothesis video segmentation from superpixel flows. In *European conference on Computer vision*, pages 268–281. Springer, 2010.

[41] Haw-Shiuan Chang and Yu-Chiang Frank Wang. Superpixel-based large displacement optical flow. In *2013 IEEE International Conference on Image Processing*, pages 3835–3839. IEEE, 2013.

[42] Deqing Sun, Ce Liu, and Hanspeter Pfister. Local layering for joint motion estimation and occlusion detection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1098–1105. IEEE, 2014.

[43] Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.

[44] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, Aug 1995.

[45] Samuel Blackman and Robert Popoli. Design and analysis of modern tracking systems(book). *Norwood, MA: Artech House, 1999.*, 1999.

[46] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[47] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[48] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[49] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 900–906. IEEE, 2003.

[50] Alexander T Ihler and David A McAllester. Particle belief propagation. In *AISTATS*, pages 256–263, 2009.

[51] James Coughlan. A tutorial introduction to belief propagation. *The Smith-Kettlewell Eye Research Institute*, 2009.

[52] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Academic Press, 2006.

[53] Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu. Large displacement optical flow from nearest neighbor fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2450, 2013.

[54] Li Xu, Jiaya Jia, and Yasuyuki Matsushita. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1744–1757, 2012.

[55] Yinlin Hu, Rui Song, Yunsong Li, Peng Rao, and Yangli Wang. Highly accurate optical flow estimation on superpixel tree. *Image and Vision Computing*, 52:167–177, 2016.

[56] Li Xu, Jianing Chen, and Jiaya Jia. A segmentation based variational model for accurate optical flow estimation. In *European Conference on Computer Vision*, pages 671–684. Springer, 2008.

[57] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jimenez Cardenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3527–3534, 2013.

[58] Oren Freifeld, Yixin Li, and John W Fisher. A fast method for inferring high-quality simply-connected superpixels. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2184–2188. IEEE, 2015.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: İnce, Kutalmış Gökalp
Nationality: Turkish (TC)
Date and Place of Birth: 06.01.1984, Bornova
E-mail: kutalmisince@gmail.com
Phone: 0 532 4741326

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Electrical and Electronics Engineering, METU	2009
B.S.	Electrical and Electronics Engineering, METU	2006
High School	Hüseyin Yalçın Çapan Lisesi	2001

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2006-current	ASELSAN INC.	System Engineer

PUBLICATIONS

Ince K.G., Cigla C., Alatan A.A.; Local Adaptive Super-Pixels, International Conference on Image Processing, IEEE, 2015.

Yildirim A., Alkar A.Z., Ince K.G.; Background modelling for pan tilt cameras, Sig-

nal Processing and Communications Applications Conference (SIU), IEEE, 2014. p. 1191-1194.

Mencguc F., Erdener A., Ari E.O., Ataseven Y., Deniz B., Ince K.G., Kazancioglu U.; Team Cappadocia Design for MAGIC 2010 (The ASELSAN Team), Land Warfare Conference 2010.