Prior-free Dependent Motion Segmentation using Helmholtz-Hodge Decomposition based Object-Motion Oriented Map

Cui-Cui Zhang[1], Zhi-Lei Liu[2]*, Member, CCF

[1]*School of Marine Science and Technology, Tianjin University, Tianjin, 300072, China*
[2]*Tianjin Key Laboratory of Cognitive Computing and Application, School of Computer Science and Technology, Tianjin University, Tianjin, 300072, China*

E-mail: cuicui.zhang@tju.edu.cn, zhileiliu@tju.edu.cn

Revised April 7th, 2017

Abstract Motion segmentation in moving camera videos is a very challenging task because of the motion dependence between the camera and moving objects. Camera motion compensation is recognized as an effective approach. However, existing work depends on prior-knowledge on the camera motion and scene structure for model selection. This is not always available in practice. Moreover, the image plane motion suffers from depth variations, which leads to depth-dependent motion segmentation in 3D scenes. To solve these problems, this paper develops a prior-free dependent motion segmentation algorithm by introducing a modified Helmholtz-Hodge decomposition (HHD) based object-motion oriented map (OOM). By decomposing the image motion (optical flow) into a curl-free and a divergence-free component, all kinds of camera-induced image motions can be represented by these two components in an invariant way. And the HHD identifies the camera-induced image motion as one segment irrespective of depth variations with the help of OOM. To segment object motions from the scene, we deploy a novel spatio-temporal constrained Quadtree labeling. Extensive experimental results on benchmarks demonstrate that our method improves the performance of state-of-the-art by 10%-20% even over challenging scenes with complex background.

Keywords Prior-free dependent motion segmentation, Helmholtz-Hodge decomposition, object-motion oriented map, Quadtree labeling

1 Introduction

Motion segmentation plays a central role in video analysis, such as coding, content-based retrieval, surveillance, and so on [1]. Extensive studies have been done on stationary camera scenarios. Most recently, more and more attentions have been paid to the dynamic scenarios taken by a moving camera (e.g., handhold camera), where scene objects are moving relative to the camera producing a dependent motion filed [2]. In many applications, e.g., video stabilization, structure-from-motion, and video editing, the camera-induced image motion (inlier) plays a more important role [3, 4]. But in other applications, e.g., action recognition and content-based retrieval, the local object motions (outliers) attract more attentions in the previous research [5]. This paper tries to distiguish the difference between inliers and outliers, and to recover them

*This is the corresponding author. This paper is recommend by CVM 2017. This work is supported by National Natural Science Foundation of China under Grant No. 61503277.

into their true values for higher level applications, which is referred to as “dependent motion segmentation”.

1.1 Problem Review and Related Work

Dependent motion segmentation in dynamic scenes is a very challenging task due to the unconstrained and unforeseen camera motion, which may cause the changes of all pixels of an image. The observed 2D image motion is caused by the 3D motions of both moving objects and moving cameras with internal camera parameters (e.g., camera zoom). These mixed and interdependent effects make this study more obscure.

Due to the fact that local object motions are very complicated, it is difficult to model and segment them directly. An effective approach is to compensate the camera-induce image motion first, and regard the residue motions to be motions of moving objects. So, we need to find an appropriate way to compensate camera-induced image motion at first. A common theme in representing motions is either by trajectories of key features or by dense motion field (optical flow). Accordingly, prior methods in the literature can be broadly divided into: (1) feature based approaches and (2) dense motion field based approaches.

Feature based methods focus on the classification of trajectories of selected features into different groups (subspaces) [6, 7]. Representative methods include: factorization-based, algebraic, and statistical based methods. Factorization-based methods [8] attempt to directly factor the trajectory matrix of multiple motions into submatrices of different independent motions, and cannot deal well with dependent motions. Algebraic methods, e.g., Generalized Principal Component Analysis (GPCA) [9], can partially deal with dependent motions. A representative statistical method named Multi-Stage Learning (MSL) was proposed in [10], which is proposed based on Costeira and Kanade’s factorization method (CK) [11] and Kanatani’s subspace separation method (SS) [12]. Several cases of camera motions can be well handled through feature based methods, in which, a simplified orthographic camera model similar to perspective camera is often assumed, and the focal length of a camera is required to be long enough to avoid any perspective distortions on the depth of 3D points. However, these requirments cannot be staisified in many practical applications. Moreover, these methods rely much on the robust feature extraction and tracking algorithm to obtain the key trajectory features of motions [13, 14]. In addition, as they only output a segmentation with sparse key features, postprocessing is necessary to obtain a dense segmentation.

Dense motion field based methods perform pixel-level segmentation on image plane motion. They usually assume the camera-induced image motion by a parametric transformation ranging from translation to perspective transformation using different parameters [15]. Pixels that are consistent with the estimated model are supposed to be inliers, while others are supposed to be outliers. Since the inlier estimation is often affected by outliers, several outlier removal methods were proposed. For example, a regression scheme, using gradient descent (GD) [15] or least squares (LS) [16], was applied to refine the inlier model by iteratively excluding outliers. Outlier rejection filter [17] explicitly filtered motion vectors by checking their similarity in a predefined window. RANSAC [18] is a statistical method, which estimates the inlier on the data containing outliers by iteratively updating the probability of inlier. Recently, a joint inlier estimation and motion segmentation method was proposed in [19, 20], which performs inlier estimation and outlier rejection simultaneously. In contrast to feature based methods, dense motion field based methods do not rely on the strong key features tracked throughout the scene. However, they suffer from two big problems: (1) The parametric models used for inlier estimation are just approximations of camera motions, which is only appropriate for the restricted cases of camera motion and scene structure. Moreover, the prior-knowledge for model selection is also required.

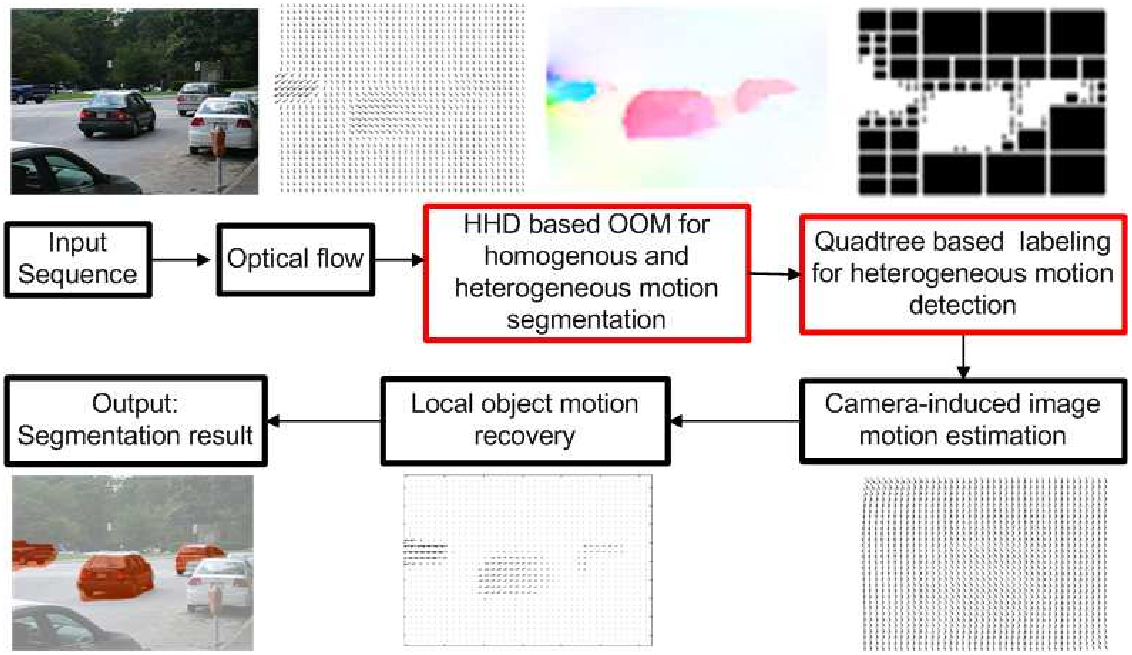

Input Sequence
Optical flow
HHD based OOM for homogenous and heterogeneous motion segmentation
Quadtree based labeling for heterogeneous motion detection
Camera-induced image motion estimation
Local object motion recovery
Output: Segmentation result

Fig. 1. Prior-free dependent motion segmentation using Helmholtz-Hodge decomposition (HHD) based object-motion oriented map (OOM).

(2) The image plane motion depends too much on the distance of 3D points from the camera. Objects at different depths may have different optical flows even if they share the same real-world motion, which leads to depth-dependent segmentation. To overcome this problem, a state-of-the-art [21] was proposed using utilized optical flow direction, other than magnitude for coherent motion segmentation. However, it only works well for camera translation, camera zoom and rotation are still challenges for this method.

Reviewing those drawbacks of existing methods, we can conclude that the main challenges in dependent motion segmentation is to find an appropriate way to represent the camera-induced image motions, which should be appropriate for all kinds of camera motions (translation, zoom in/out, or rotation) without any prior-knowledge and restrictions on the camera motion and scence structure. So, as the appearance based segmentation suffers from depth variations in 3D scenes, we should also develop new algorithms dealing effectively with depth variations.

1.2 Overview of Our Approach

In this paper, a prior-free dependent motion segmentation algorithm is proposed by introducing a Helmholtz-Hodge decomposition (HHD) based object-motion oriented map (OOM). HHD decomposes the image motion into a curl-free and a divergence-free component. Without any prior-knowledge on the camera motion and scene structure, any kinds of camera-induced image motions can be represented by these two components in an invariant way. The outline of our algorithm is illustrated as Fig. 1.

The modified HHD can also make the segmentation independent of depth variations by adding two assumptions to the original HHD [22, 23]. It identifies the inlier as one segment irrespective of depth variations in 3D scenes at first. Then, the heterogeneous motion caused by depth discontinuities and moving objects will be computed and processed further. In this paper, an OOM will be constructed to detect such heterogeneous motion. As the heterogeneous motion varies at different locations on OOM, it is difficult to label them based on global thresholding. In this paper, a Spatio-Temporal constrained Quadtree labeling method will be introduced. HHD based OOM will be utlized to separate the homogeneous motions and the heterogeneous motions. To compensate depth discontinuity, a low-order polynomial based surface fitting method is employed for inlier estimation followed by the outlier recovery and segmentation.

Unlike existing methods, our proposed method does not rely on prior-knowledge for camera motion modeling. All kinds of camera-induced image motions can be represented by our algorithm in an invariant way. Moreover, our algorithm is robust to depth variations, providing assurance on high ability of our method in coping with real-world scenes.

1.3 Contributions

The contributions of our work are as follows:

- Without any prior-knowledge on the camera motion and scene structure, HHD interprets all kinds of camera-induced image motions by its two components in an invariant way.
- The modified HHD identifies the camera-induced image motion as one segment irrespective of depth variations and records heterogeneous motions on OOM.
- A spatio-temporal constrained Quadtree labeling method is proposed for the separation of heterogeneous motions from OOM.

The rest of this paper is organized as follows: Section 2 describes motion flow modeling in 2D/3D scenes and inlier/outlier representations based on optical flow estimation. Section 3 introduces the modified HHD and the interpretation of camera-induced image motion by HHD. OOM is constructed in Section 4 with a Quadtree based labeling. Section 5 presents the inlier estimation and outlier recovery. Experiments and disscussions are shown in Section 6. Finally, we conclude this work in Section 7.

2 Motion Flow Modeling

2.1 Motion Representation

Camera undergoes two kinds of 3D motions: translation $\boldsymbol{T} = (T_X, T_Y, T_Z)$ and rotation $\boldsymbol{R} = (R_X, R_Y, R_Z)$. The instantaneous image motion of a general 3D scene can be formulated as:

$$\begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} -f_c(\frac{T_X}{Z} + R_Y) + x\frac{T_Z}{Z} + yR_Z - x^2\frac{R_Y}{f_c} + xy\frac{R_X}{f_c} \\ -f_c(\frac{T_Y}{Z} + R_X) - xR_Z + y\frac{T_Z}{Z} - xy\frac{R_Y}{f_c} + y^2\frac{R_X}{f_c} \end{pmatrix}, \tag{1}$$

where f_c is the focal length, Z is the depth of the 3D point, and $(u(x,y), v(x,y))$ denotes the image plane motion at coordinate (x,y). We can see that the length of a motion vector is inversely proportional to its depth. Image motion has more specific representations in different scenes (2D or 3D).

2.1.1 *Motion in 2D Scene*

When the scene is parallel to the image plane, all the scene objects are at the same distance from the camera without depth variations ($\Delta Z = 0$). The camera-induced image motion can be modeled by a single parametric transformation, which is of low-order polynomial function:

$$\begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} a_1 + a_2x + a_3y + a_4x^2 + a_5xy \\ a_6 + a_7x + a_8y + a_4xy + a_5y^2 \end{pmatrix},$$

where the parameters $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)$ are functions of camera motion $(\boldsymbol{R}, \boldsymbol{T})$, f_c, and Z as depicted in Eq.(1). Here, Z is constant.

The scene satisfying above-mentioned conditions is regarded as 2D scene. In practice, the scene may have small depth variations. However, when the depth variation within the scene is much smaller than the overall 3D range of the scene from the camera, these scenes can also be considered as 2D scenes [24].

2.1.2 Motion in 3D Scene

When the scene with large depth variations, it cannot be approximated by a flat/planar surface which is parallel to the image motion. In this case, we need to use a set of planar surfaces instead of a single one to represent it. We define such scenes as 3D scenes. Camera motions under 3D scenes can be considered in the following cases:

- Camera translation: When the camera is translating, the image motion in Eq.(1) becomes:

  $$\begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} -\frac{T_X}{Z} + x\frac{T_Z}{Z} \\ -\frac{T_Y}{Z} + y\frac{T_Z}{Z} \end{pmatrix},$$

  Camera translation consists of two cases: (1) translation, which is parallel to the image plane, so $T_Z = 0$; and (2) camera zoom in/out, where $T_X = 0, T_Y = 0$. Camera zoom contributes to a radial image motion (expansion or contraction). For both cases, the image motion observed in the scene depends on the depth Z of scene points. Different planar surfaces have substantially different induced image motions depending on their depths. A single parametric transformation becomes insufficient for modeling the image motion. We need to use a set of parametric transformations $\{P_1, P_2, ..., P_N\}$ to represent it, where N denotes the number of planar surfaces.

- Camera Rotation: When the camera undergoes a pure rotation, Eq. (1) becomes:

  $$\begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} -R_Y + yR_Z - x^2 R_Y + xyR_X \\ R_X - xR_Z - xy2R_Y + y^2 R_X \end{pmatrix},$$

  The contribution of camera rotation to the image plane motion is independent of scene point's depth Z.

Any kinds of camera-induced image motion $\boldsymbol{C}_{IM}$ can be regarded as the combination of three basic motions: translation $\boldsymbol{T}_{IM}$, radial motion (or named as perspective motion) $\boldsymbol{P}_{IM}$, and rotation $\boldsymbol{R}_{IM}$ by:

$$\boldsymbol{C}_{IM} = \alpha \boldsymbol{T}_{IM} + \beta \boldsymbol{P}_{IM} + \gamma \boldsymbol{R}_{IM}, \qquad (2)$$

where α, β, γ are three regularization parameters.

Among these three basic motions, translation and radial motion are influenced by depth variations in 3D scenes. Although rotation is independent of depth validations, another problem should be considered: the intuitive interpretation of 2D motion field is generally based on Cartesian coordinate system with two bases x, y. A motion vector can be projected into x and y components, denoted by u and v, respectively. For camera translation, we can use an invariant parameter $\frac{u}{v}$ to interpret it. But for the rotation and radial motion, $\frac{u}{v}$ changes with x, y changing on motion field, which cannot be represented by an invariant parameter using existing methods.

2.2 Inlier/Outlier Representation

In this paper, inlier is the abbreviation of inlier optical flow, which refers to the image motion of static scene object caused by camera motion. Outlier is the abbreviation of outlier optical flow, which refers to the image motion of moving object caused by both camera motion and object motion. Existing optical flow estimation methods usually assume a perspective camera model other than a simplified orthographic camera mode [25]. Compared to the orthographic camera model, the perspective model is more robust to perspective distortions. It does not require the camera focal length to be long enough to avoid perspective distortions. The optical flow field under perspective projection is in Eq.3 as follows:

$$\begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} f(\frac{T_x}{z} + \Omega_2) - \frac{T_z}{z}x - \Omega_3 y - \frac{\Omega_1}{f}xy + \frac{\Omega_2}{f}x^2 \\ f(\frac{T_y}{z} - \Omega_1) - \Omega_3 x - \frac{T_z}{z}y + \frac{\Omega_2}{f}xy - \frac{\Omega_1}{f}y^2 \end{pmatrix}, \qquad (3)$$

where f is the focal length, $\boldsymbol{\Omega} = [\Omega_1, \Omega_2, \Omega_3]$ is the angular velocity vector, and $\boldsymbol{T} = [T_x, T_y, T_z]$ is the translation velocity vector. We can see that the optical flow field under camera translation is dependent on the depth Z of 3D points. It is incompressible in the 2D motion field.

Optical flow estimators often add some constraints to make the estimation robust and accurate. Commonly used constraints include the intensity constancy assumption (interpreted by an energy function $E_{\text{Intensity}}$), gradient constancy assumption (interpreted by E_{Gradient}), and local smoothness assumption (interpreted by $E_{\text{Smoothness}}$). The optical flow field is obtained by minimizing the total energy: $E = E_{\text{Intensity}} + E_{\text{Gradient}} + E_{\text{Smoothness}}$. Benefiting from local smoothness constraint, for 2D scenes with small depth variations, the optical flow field is piecewise smooth and can be represented by a single parametric transformation of low-order polynomial function (P^L). But for 3D scenes with large depth variations, the local smoothness constraint fails and the estimation becomes inaccurate. We should use a high-order polynomial function (P^H) to represent it as following:

$$\begin{aligned} \boldsymbol{OF}_{2D} &= P_1 = P^L \\ \boldsymbol{OF}_{3D} &= \{P_1, P_2, ..., P_N\} \approx P^H, \end{aligned}$$

Most part of the inlier belongs to the inner part of static scene objects, which are with small depth variations. Their motion fields are "homogeneous". Here, the "homogeneous" implies the continuity of the motion field. However, motion discontinuities occur at the boundaries of static scene objects due to depth discontinuities. Their motion fields are heterogeneous. Outliers, which belong to the moving objects, also suffer from motion discontinuities due to the relative motion between camera and moving objects. Both the inlier and outlier can be represented by the homogeneous part and heterogeneous part as below:

$$\begin{aligned} \boldsymbol{V}_{inlier} &= a_1 \boldsymbol{V}_{inlier}^{homo.} + b_1 \boldsymbol{V}_{inlier}^{hetero.}, a_1 >> b_1, \\ \boldsymbol{V}_{outlier} &= a_2 \boldsymbol{V}_{outlier}^{homo.} + b_2 \boldsymbol{V}_{outlier}^{hetero.}, a_2 << b_2. \end{aligned}$$

Most part of inlier is homogeneous, thus $a_1 >> b_1$, which is opposite for the outliers, of which $a_2 << b_2$. The homogeneous part should be represented by a low-order polynomial function, while the heterogeneous part should be represented by a high-order polynomial function.

$$\boldsymbol{OF}_{3D} = \{P_1, P_2, ..., P_N\} \approx P^H \tag{4}$$

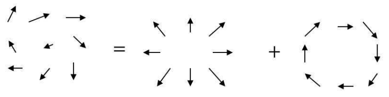
3 Camera-Induced Image Motion Representation & HHD

3.1 Principle of HHD

As these three basic image motions have variant interpretations under different camera motions and scene structures, they cannot be represented by a unified form using existing methods. So, prior-knowledge is required for model selection. Moreover, we cannot find an appropriate interpretation to represent rotation and radial motion based on Cartesian coordinate system using existing methods. Due to these facts, an invariant motion representation algorithm is proposed based on Helmholtz-Hodge decomposition (HHD), which represents variant camera-induced image motions in a uniform way and characterizes rotation and radial motion by its curl-div regularizers effectively.

HHD is one of the fundamental theorems in fluid dynamics. Theoretically, it decomposes an arbitrary flow field $\boldsymbol{\xi}$ to two components: a curl-free component $\nabla \boldsymbol{E}$ and a divergence-free component $\nabla \times \boldsymbol{W}$:

$$\boldsymbol{\xi} = \nabla \boldsymbol{E} + \nabla \times \boldsymbol{W}.$$



Here, $\boldsymbol{E}$ and $\boldsymbol{W}$ are 3D potential surfaces defined as: (1) **Scalar potential surface**, whose gradient is the curl-free component $\nabla \boldsymbol{E}$; (2) **Vector**

potential surface, whose curl operation denotes the divergence-free component $\nabla \times \boldsymbol{W}$. The curl operator and the gradient operator have the following relationship:

$$\nabla \times \boldsymbol{W} = (\nabla \boldsymbol{W})^{\perp}. \tag{5}$$

3.2 Camera Motion Factorization Using HHD

Based on the principle of HHD explained in [26], the motion of a volume element in the three dimensional space consists of three kinds of motions: (1) expansion or contraction(which is also named as radial motion), (2) rotation, and (3) translation. The expansion or contraction exists only in the curl-free component due to their irrotationality. Similarly, the rotation exits only in the divergence-free component because of its incompressibility. However, translation can exist both in the curl-free component and divergence-free component because of its incompressibility and irrotationality.

Thus, as depicted in Eq.(2), any camera motion consisting of these three kinds of motions (radial motion, rotation, and translation) can be represented by these two components of HHD as follows:

- **Camera Radial Motion**: it is irrotational and is present in the curl-free component only. That is:

$$\boldsymbol{P}_{IM} = \nabla \boldsymbol{E}.$$

- **Camera Rotation**: it is incompressible and is present in the divergence-free component only. That is:

$$\boldsymbol{R}_{IM} = \nabla \times \boldsymbol{W}.$$

- **Camera Translation**: it is both incompressible and irrotational and can be present in both components equally. That is:

$$\boldsymbol{T}_{IM} = \frac{1}{2}\nabla \boldsymbol{E} + \frac{1}{2}\nabla \times \boldsymbol{W}.$$

Please refer to [26, 27] for more details.

3.3 Implementation of HHD

In previous research, many algorithms have been proposed for the implementation of HHD. For example, Polthier et al. [23] derived a technique for 2D discrete vector fields. Tong et al. [22] extended it to discrete vector fields on 3D meshes. To ensure HHD be able to identify the inlier as one segment irrespective of depth variations, we add two assumptions to the previously defined HHD and introduce an modified HHD in this paper. These two assumptions are: (1) the original motion field should be piece-wise smooth; (2) HHD is performed based on global minimization. The implementation is as follows: since $\nabla \boldsymbol{E}$ and $\nabla \times \boldsymbol{W}$ are the projections of original motion field $\boldsymbol{\xi}$ to the space of curl-free field and divergence-free field, respectively, the distances between $\boldsymbol{\xi}$ and two projected components should be minimal. Therefore, energy minimization is applied to calculate these two components:

$$\begin{aligned} \min(D(\boldsymbol{E})) &= \min(\textstyle\int_{\boldsymbol{\Omega}} \parallel \nabla \boldsymbol{E} - \boldsymbol{\xi} \parallel^2 d\boldsymbol{\Omega}), \\ \min(G(\boldsymbol{W})) &= \min(\textstyle\int_{\boldsymbol{\Omega}} \parallel \nabla \times \boldsymbol{W} - \boldsymbol{\xi} \parallel^2 d\boldsymbol{\Omega}), \end{aligned} \tag{6}$$

where $\boldsymbol{\Omega}$ denotes the image domain. According to the definition of HHD, the divergence-free component ($\nabla \times \boldsymbol{W}$) does not exist in the curl-free component ($\nabla \boldsymbol{E}$), and vice versa. We can derive the following criteria:

$$\begin{aligned} \textstyle\int_{\boldsymbol{\Omega}} \nabla \times (\nabla \boldsymbol{E}) d\boldsymbol{\Omega} = \textstyle\int_{\boldsymbol{\Omega}} \nabla \times (\boldsymbol{\xi} - \nabla \times \boldsymbol{W}) d\boldsymbol{\Omega} = 0, \\ \textstyle\int_{\boldsymbol{\Omega}} \nabla \cdot (\nabla \times \boldsymbol{W}) d\boldsymbol{\Omega} = \textstyle\int_{\boldsymbol{\Omega}} \nabla \cdot (\boldsymbol{\xi} - \nabla \boldsymbol{E}) d\boldsymbol{\Omega} = 0. \end{aligned} \tag{7}$$

In the discrete domain, Eq.(7) can be rewritten as:

$$\begin{aligned} \textstyle\sum_{i \in \boldsymbol{\Omega}} \nabla \times (\nabla \times \boldsymbol{W}_i) = \textstyle\sum_{i \in \boldsymbol{\Omega}} \nabla \times \boldsymbol{\xi}_i, \\ \textstyle\sum_{i \in \boldsymbol{\Omega}} \nabla \cdot (\nabla \boldsymbol{E}_i) = \textstyle\sum_{i \in \boldsymbol{\Omega}} \nabla \boldsymbol{\xi}_i. \end{aligned} \tag{8}$$

Since they are linear functions, we can abbreviate them as:

$$\boldsymbol{S}_1 \boldsymbol{E} = \boldsymbol{B}, \qquad \boldsymbol{S}_2 \boldsymbol{W} = \boldsymbol{C}. \tag{9}$$

where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are $N \times N$ sparse element matrices, $\boldsymbol{E}$ and $\boldsymbol{W}$ are $N \times 1$ vectors to be calculated, $\boldsymbol{B}$ and $\boldsymbol{C}$ represent the right side of Eq.(8). The potential surfaces $\boldsymbol{E}$ and $\boldsymbol{W}$ are calculated by solving Eq.(9). Then $\nabla \times \boldsymbol{W}$ is subsequently obtained by Eq.(5).

3.4 Inlier and Outlier Representation by HHD

Based on the previous discussion (see Eq.(4)), the optical flow field can be represented by a low-order polynomial function for $\boldsymbol{V}^{\text{homo.}}_{\text{inlier}}$, $\boldsymbol{V}^{\text{homo.}}_{\text{outlier}}$, and by a high-order polynomial function for $\boldsymbol{V}^{\text{hetero.}}_{\text{inlier}}$, $\boldsymbol{V}^{\text{hetero.}}_{\text{outlier}}$. In our algorithm, the polynomial function actually corresponds to the potential surface of HHD. As we added two assumptions to the implementation of HHD, the potential surfaces of the modified HHD are piece-wise smooth. They approximate the basic shape of motion field and thus correspond to a low-order polynomial function. Thus, the homogeneous motion field, which should be represented by a low-order polynomial function, can be interpreted by two components of HHD as follows:

$$\begin{aligned} \boldsymbol{OF} = \boldsymbol{V}^{\text{homo.}}_{\text{inlier}} + \boldsymbol{V}^{\text{hetero.}}_{\text{inlier}} + \boldsymbol{V}^{\text{homo.}}_{\text{outlier}} + \\ \boldsymbol{V}^{\text{hetero.}}_{\text{outlier}}, \\ \{\boldsymbol{V}^{\text{homo.}}_{\text{inlier}}, \boldsymbol{V}^{\text{homo.}}_{\text{outlier}}\} \Rightarrow P^L \Rightarrow HHD, \\ = k_1(\nabla \boldsymbol{E}) + k_2(\nabla \times \boldsymbol{W}), \end{aligned} \tag{10}$$

where k_1 and k_2 are two regularization parameters, which will be determined in Section 4.

The heterogeneous motion corresponding to a high-order polynomial function cannot be represented by HHD will be computed and processed further.

4 Object-Motion Oriented Map Construction & Heterogeneous Motion Labeling

4.1 Object-Motion Oriented Map (OOM)

To construct OOM, the homogeneous motion will be calculated according to Eq. (10) at first. Then, OOM will be obtained by subtracting the homogeneous motion from the original motion field $\boldsymbol{\xi}$. The key here is the determination of the two parameters k_1 and k_2 according to the type of the camera motion involved in the scene. In this paper, two distance functions in Eq.(11) are defined for this purpose.

$$\begin{aligned} d_1 &= \sum(\frac{\|\boldsymbol{\xi} - \nabla \boldsymbol{E}\|}{\|\boldsymbol{\xi}\|}), \\ d_2 &= \sum(\frac{\|\boldsymbol{\xi} - \nabla \times \boldsymbol{W}\|}{\|\boldsymbol{\xi}\|}), \end{aligned} \tag{11}$$

where d_1 denotes the distance between $\boldsymbol{\xi}$ and the curl-free component, and d_2 denotes the distance between $\boldsymbol{\xi}$ and the divergence-free component. As aforementioned, radial motion only exists in the curl-free component, rotation only exits in the divergence-free component, and translation can present in both components. We have the following three observations:

- If $d_1 < 0.5$, and $d_2 > 0.5$, the curl-free component is much similar to the original optical flow field, which is different from the divergence-free component. In addition, the camera-induced image motion belongs to a radial motion, and exists only in the curl-free component. So, $k_1 = 1$, and $k_2 = 0$.
- If $d_1 > 0.5$, and $d_2 < 0.5$, the divergence-free component is much similar to the original optical flow field, which is different from the curl-free component. That is, camera-induced image motion belongs to rotation, and exists only in the divergence-free component. So, $k_1 = 0$, and $k_2 = 1$.
- If $d_1 < 0.5$, and $d_2 < 0.5$, both components are similar to the original motion field. The camera-induced image motion is translation and exists in both components. So, $k_1 = k_2 = 1/2$.

As camera motion is a generic motion, all the scene points should share the same kind of camera motion. Thus, k_1 and k_2 are invariant parameters across the whole scene. We can use a unique k_1

and k_2 for all the pixels of the scene. After determining k_1, k_2, OOM can be calculated as:

$$\boldsymbol{OOM} = \boldsymbol{\xi} - k_1(\nabla \boldsymbol{E}) - k_2(\nabla \times \boldsymbol{W}),$$

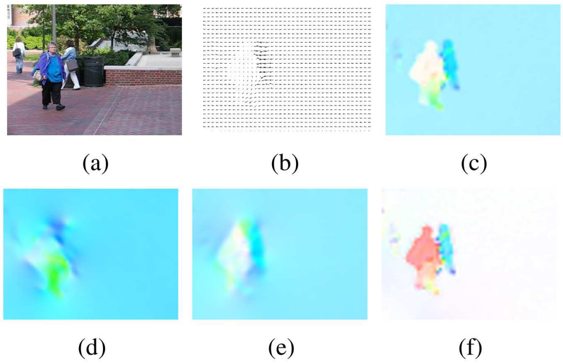


Fig. 2. An example of 3D scene. (a) One frame. (b) The input optical flow. (c) Visualization of the input optical flow by color coding [28]. (d) Curl-free component. (e) Divergence-free component. (f) OOM

An example of 3D scene is shown in Fig. 2 to illustrate the procedure of OOM construction. In Fig. 2(a), two persons are walking from left to right, and a camera is also moving from left to right. From the original optical flow field in Fig. 2(b) and its color image in Fig. 2(c), we can find that it is difficult to segment these two persons from the scene directly. After HHD decomposition, the OOM in Fig. 2(f) demonstrates that most part of homogeneous motion field of static scene objects has been removed from OOM, and only heterogeneous motion caused by depth discontinuities and moving objects are left.

4.2 Spatio-Temporal Constrained Quadtree Labeling

To detect the heterogeneous motions, we need to label them from OOM at first. As depth discontinuities and moving objects vary at different locations, it is very difficult to label them based on global thresholding. To this end, a data-driven Quadtree scheme casting the heterogeneous motion labeling to local subregions is proposed in this paper. If a region is determined as heterogeneous according to a criterion function in Eq.(12), it will be further divided into four subregions. For the Quadtree labeling, one of the most important thing is to define the criterion function for partition. Without considering the temporal constraint, the partition can only be defined in the spatial domain frame by frame. Following two conditions should be considered in this criterion function as presented in [29, 30]:

- The variance of a region R should be higher than or equal to a threshold variance T_{var};
- The mean value of R should be higher than or equal to a threshold mean value T_{mean};

And the mathematical depiction is:

$$doSplit(R) = true, \text{while} \begin{cases} \text{var}(R) \geq T_{\text{var}} & or, \\ \text{mean}(R) \geq T_{\text{mean}}. \end{cases} \quad (12)$$

The first condition is utlized to detect the heterogeneous motion caused by depth discontinuities ($\boldsymbol{V}^{\text{hetero.}}_{\text{inlier}}$), in which, the value changes violently in local regions, and the second condition is utlized to detect the heterogeneous motion caused by moving objects ($\boldsymbol{V}^{\text{hetero.}}_{\text{outlier}}$), in which, the local object motions occupy larger part of the region if the value is higher than the threshold. Partition is performed untill no more regions can be split. Finally, regions with smallest size $smallR$ are labeled as foreground regions containing heterogeneous motions, and will be excluded from the inlier estimation in the next procedure. The rest larger regions $largeR = wholeR - smallR$ are regared as homogeneous region and will be evolved in inlier estimation, where $wholeR$ represents the whole region of OOM.

However, there are still some problems as shown in Fig. 3, the segmentation results of a video may be unsmooth along the time dimension. That is, the segmentation results of the previous frame may largely differ from that of the current frame. To solve this problem, Quadtree labeling is performed by introducing a spatio-temporal constraint in this paper. In addition, the previous partition result will be utlized as the initial condition of the current frame. The criterion function consists of three considerations as following:

- The partition result of region R at the current frame equals to the partition result of the previous frame;
- The partition result of region R will be false if its variance is lower than or equal to the value of threshold variance $1 - T_{\text{var}}$;
- The partition result of region R will be false if its mean is lower than or equal to the value of threshold mean $1 - T_{\text{mean}}$;

And the mathematical representation is:

$$doSplit(R_{\text{current}}) = doSplit(R_{\text{pre}}),$$

$$doSplit(R) = false, while \begin{cases} var(R) \leq 1 - T_{\text{var}} \quad or, \\ mean(R) \leq 1 - T_{\text{mean}}. \end{cases}$$

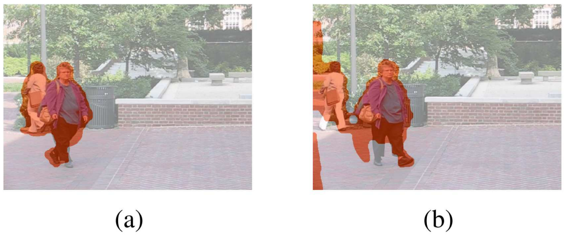

(a) (b)

Fig. 3. Segmentation results on two consistent frames of People 2 video sequence. (a) Frame 3. (b) Frame 4.

Usually, OOM is utlized to split the motion into small number of larger blocks when the thresholds T_{var}, T_{mean} are large, and vice versa. Large threshold value indicates that not all heterogeneous motions are well detected, and small threshold value indicates heterogeneous motions are over-segmented. By defining a universal threshold, we may suffer from such under-segmentation or over-segmentation problems. To avoid this, a data-adaptive threshold definition algorithm is proposed based on the Quadtree structure, which is described as follows.

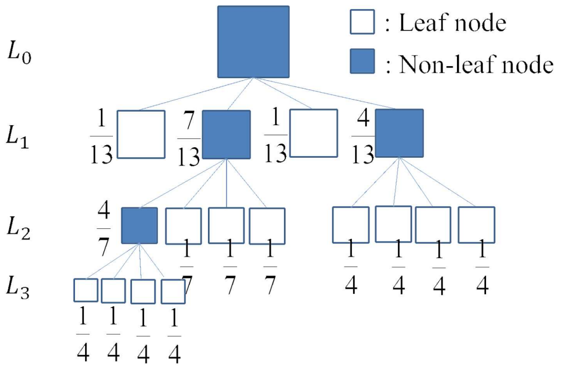
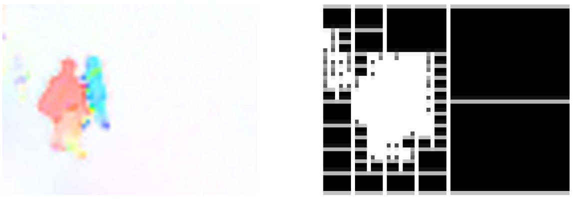


Fig. 4. An example Quadtree partition.

1. Initial estimations are assigned for the thresholds: $T_{\text{var}} = var(wholeR)$, $T_{\text{mean}} = mean(wholeR)$.
2. A hierarchical Quadtree structure containing leaf and non-leaf nodes will be obtained based on OOM partition using T_{var} and T_{mean} of Eq.(12). As shown in Fig. 4, the variance $var(subR_i)$ and mean value $mean(subR_i)$ can be calculated for each leaf node $subR_i$.
3. The variance and mean values of each non-leaf node $subR_j$ can be obtained by averaging its four children based on a weighted sum rule. The weight of each child is determined by dividing the number of its leaf nodes by the leaf number of all four children. A higher weight implies a subregion is foreground with higher probability. Take Fig. 4 as an example, there are four nodes at level L_1, which have 1, 7, 1, 4 leaf nodes, receptively. Their weights are thus to be $\frac{1}{13}, \frac{7}{13}, \frac{1}{13}, \frac{4}{13}$. When we reach the first level, the variance and mean values of the root node $var(root)$ and $mean(root)$ can be calculated, and assign them with new thresholds T'_{var} and T'_{mean}, respectively.
4. Repeat steps 2 and 3 until the distance between two adjacent Quadtrees is smaller than a predefined tolerance error ε. Each Quadtree is encoded by a 0-1 sequence, in which, 0 represents non-leaf node and 1 represents leaf node. The distance d between

two adjacent Quadtrees d is defined by dividing the Hamming distance of their 0-1 sequences by the length of the sequence. The whole procedure stops at $d < \varepsilon$. In this paper, we define $\varepsilon = 5\%$.

The thresholds will be updated according to the last Quadtree structure in each repetition of our algorithm. The new thresholds are closer to the values of foreground regions containing heterogeneous motions. Finally, the most appropriate thresholds for each OOM will be obtained automatically. Fig. 5 shows the Quadtree partition on the example 3D scene. We can see that Quadtree is effective in labeling both moving objects and depth discontinuities from OOM.



(a) (b)

Fig. 5. Quadtree partition on the example 3D scene. (a) OOM. (b) Quadtree partition.

5 Inlier Estimation & Outliers Recovery

5.1 Inlier Estimation using Surface Fitting

After Quadtree based heterogeneous motion labeling on OOM, the remaining regions corresponding to homogeneous motion fields will be utlized for inlier estimation. The procedure will be illustrated on scalar potential surface $\boldsymbol{E}$ at first. The procedure on $\boldsymbol{W}$ is analogous.

As aforementioned, the potential surface of HHD should be smoothed and represented by a low-order polynomial function. Thus, the problem of inlier estimation from $\boldsymbol{E}$ is formulated as construction of a new smooth surface $\boldsymbol{E}'$, which approximates the smooth basic shape of $\boldsymbol{E}$. In [15], several parametric models were conducted for inlier estimation. These models are designed for camera motions ranging from simple translation to complex perspective transformations. However, prior-knowledge of motion structure is required to select an appropriate model in these method. In this paper, a prior-free solution is proposed based on surface fitting using a low-order polynomial function as follows:

$$z = a_{d0}x^{d} + a_{0d}y^{d} + \cdots + a_{ij}x^{i}y^{j} + \cdots + a_{10}x \\ + a_{01}y + a_{00},$$

The difficulty of surface fitting is how to define an appropriate degree d. It has been known that the higher the degree d, the more the details of the approximated surface will be obtained, but it will potentially evolve some local deformations. On the contrary, lower degree polynomial will yield a smoother and simpler surface approximating the inlier positions with poor accuracy. In [29, 30], a polynomial of $d = 5$ is employed to produce a smooth and accurate surface $\boldsymbol{E}'$. However, it is not appropriate for some simple surfaces, such as the surface caused by camera translation. In this case, some outliers (noises) will be included in the surface. To solve this problem, we also utilize the spatio-temporal constraint to define an appropriate degree for different kinds of camera motions. It is believed that the camera motion between two consistent frames shares the same kind of motion in our paper. And the camera rotation and radial motions are more complex than camera translation. We need to use a relatively high degree $d = 5$ to represent the surface of camera motion. For the simple camera translation, we need to use a relatively low degree $d = 3$ to represent it. The details are as follows. For the HHD decomposition of previous frame, if we find $k_1 = 1, k_2 = 0$ or $k_1 = 0, k_2 = 1$, it means this camera motion is rotation or radial motion. Then we need to assign d as 5 for the current frame. However, if $k_1 = k_2 = 1/2$, it means this camera motion is translation. We should assign $d = 3$ for this frame.

Finally, a smooth surface $\boldsymbol{E}'$ which best fits the base of $\boldsymbol{E}$ will be obtained. Similarly, we will get a new smooth potential surface $\boldsymbol{W}'$ which approximates the base of $\boldsymbol{W'}$. The inlier of curl-free component is calculated by $\boldsymbol{G}_1 = \nabla \boldsymbol{E}'$. The inlier of divergence-free component is computed by $\boldsymbol{G}_2 = \nabla \times \boldsymbol{W}'$. The final inlier optical flow is estimated by linear combination of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ using Eq.(13), where k_1, k_2 have been determined in Subsection 4.1.

$$\boldsymbol{G} = k_1\boldsymbol{G}_1 + k_2\boldsymbol{G}_2, \tag{13}$$

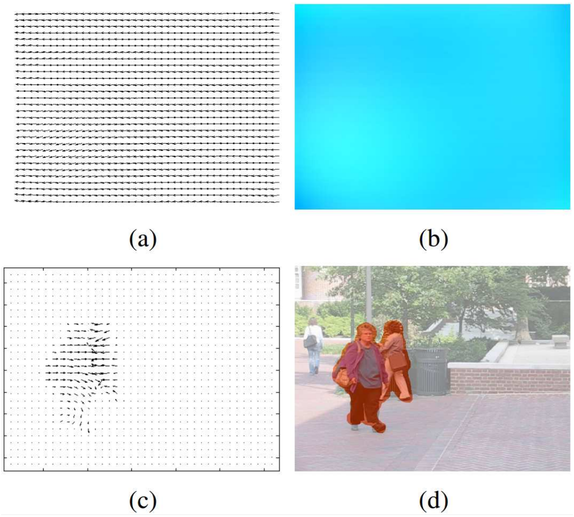


Fig. 6. Inlier estimation and outlier recovery on the example 3D scene. (a) The estimated inlier optical flow. (b) Color visualization of (a). (c) The recovered outlier optical flow. (d) Segmentation result.

Although the heterogeneous part of inlier is not involved in estimation, since most part of inlier has been involved, its heterogeneous part can be approximated by its homogeneous part. In this way, the surface fitting separates the depth discontinuities from the true moving objects. Fig. 6(a) and Fig. 6(b) present the estimated inlier of the example 3D scene. We can see that our method estimates the inlier accurately.

5.2 Outliers Recovery & Motion Segmentation

After inlier estimation, outliers can be recovered by subtracting the inlier from the original motion field subsequently. The segmentation is obtained by assigning binary labels on the pure outlier motion field. Since the surface fitting used in inlier estimation has a defect, it fits the data in the middle, but goes wild at the edge of the $x - y$ domain of the original data. To refine the segmentation map, the raw result is filtered by the mean-curvature of the original potential surfaces. Fig. 6 shows the recovered outlier motion field in Fig. 6(c) and the final segmentation map in Fig. 6(d).

6 Experiments

6.1 Datasets & Experiment Setup

The performance of our proposed method is evaluated on four benchmark datasets: Hopkins [31], Berkeley Motion Segmentation [32], Complex Background [21], and SegTrack [33]. The Hopkins dataset contains three categories of video sequence: checkerboard, car, and people, in which, the ground truth segmentation on selected features tracked throughout the sequence is also provided. Since checkerboard sequences do not correspond to natural scenes, we just use one sequence (1R2TCR) to show the effectiveness of our method in dealing with cameras rotation. The Berkeley dataset is derived from the Hopkins dataset, which consists of 26 moving camera videos of car, people, and Marple sequences. This dataset has full pixel-level annotations on multiple objects for a few frames sampled throughout the video. Since Marple sequences mainly contain static scenes or the static objects, which are not challenges of our method, this dataset is not used in our experiments. In this paper, the car and people sequences containing Hopkins and Berkeley datasets are selected to evaluate our method. In addition, another two datasets: Complex Background and SegTrack, containing extremely challenging scenes are also selected to highlight the strength of our method, in which, full pixel-level annotations on multiple objects are provided at each frame within each video.

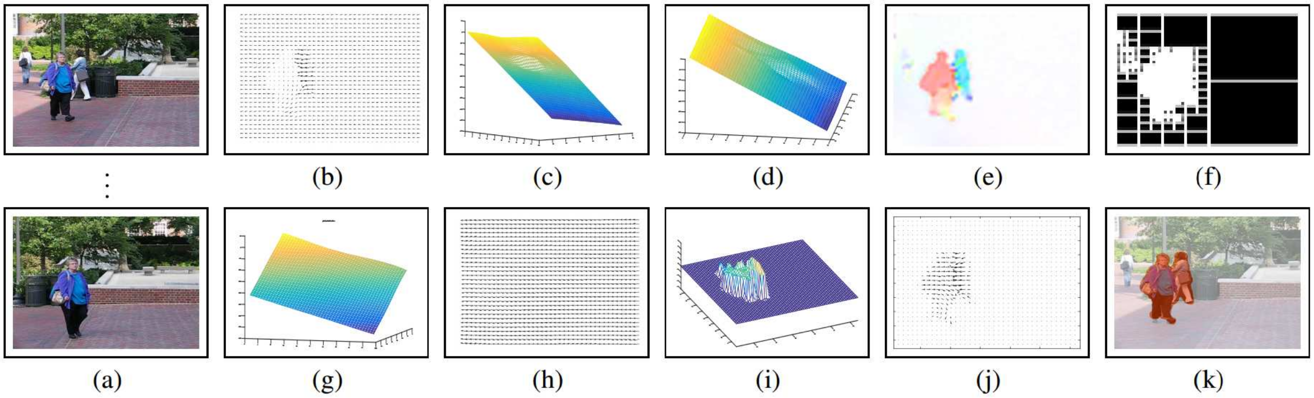


Fig. 7. Scenario 1: people2 sequence. (a) Image sequence from frame 1 to frame 30. (b) Optical flow of one frame. (c) Potential surface $\boldsymbol{E}$. (d) Potential surface $\boldsymbol{W}$. (e) The object-motion oriented map (OOM). (f) Quadtree partition on OOM. (g) The estimated inlier potential surface. (h) Inlier optical flow field. (i) The recovered local motion shown in 3D. (j) Outlier optical flow field. (k) Segmentation result.

In this subsection, three sequences are choosed to illustrate the performance of our method in dealing with different camera motions at first. Then, our method is also compared with several existing methods. Optical flow is calculated using Brox's method [25] and optimized by [34].

6.2 Performance Evaluation on Challenging Scenes

Experiments on three representative sequences: people2, 1R2TCR and parking are performed to evaluate the performance of our method in dealing with varied camera motions. We add a prefix to denote the motions involved in each scene at the begining of each sequence's name. The local objects are identified by natural numbers (e.g., 1, 2, 3, ..., N) and the camera is identified by the letter "C". The type of object motions and camera motions is indicated by following letters: "R" for rotation, "T" for translation, and "P" for radial motion. For example, if a sequence is called 1T2RCRT it means that the first object translates, the second object rotates, and the camera motion consists of both rotation and translation.

[1] 1T2TCT-people2 sequence. This sequence is from the Berkeley motion segmentation dataset, in which, two people are walking in the scene: one is from left to right, another is from right to left, and the camera is translating (see Fig. 7(a)). From the original optical flow field in Fig.7(b), we can find that it is difficult to segment these two people clearly from the scene. The potential surfaces in Fig. 7(c) and Fig. 7(d) demonstrate the homogeneous part of both inlier and outliers. In Fig. 7(e), all these two people are revealed clearly. Quadtree provides the label of heterogeneous part in Fig. 7(f). The estimated inlier potential surface in Fig. 7(g) and its motion field in Fig. 7(h) are quite smooth. Local object motions are recovered accurately in Fig. 7(i) and Fig. 7(j). This example demonstrates that our method can not only segment motions well, but also annotate them with their true values. With the help of OOM and Quadtree, our method deals with challenging scenes effectively.

[2] 1R2TCR-Checkerboard sequence. This data is from the Hopkins dataset, which involves three motions: a rotating view (inlier), in which a basket is rotating in the top left scene, and a box is translating from left to right in the bottom scene. This is also a very challenging scene, which comprises of multiple different motions.

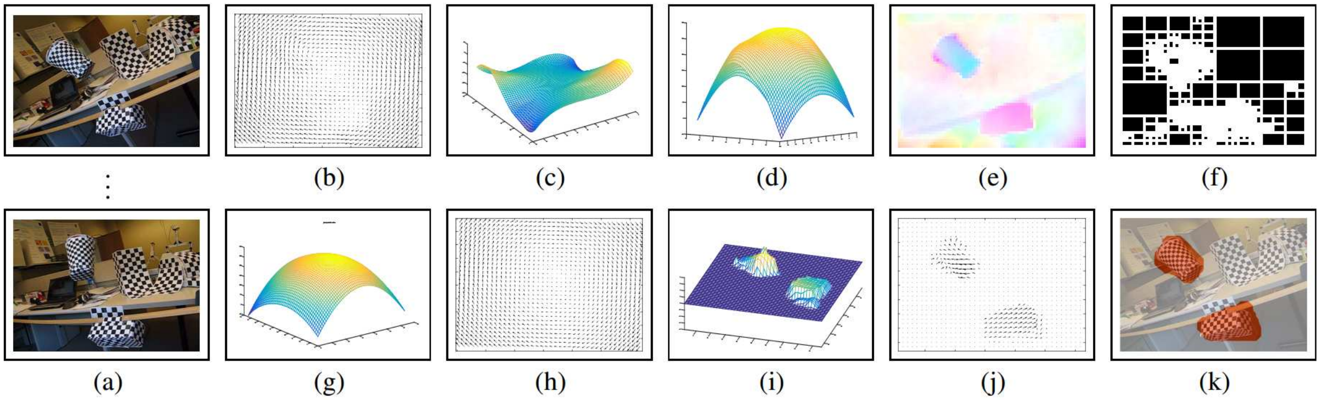


Fig. 8. Scenario 2: checkerboard sequence. (a) Image sequence from frame 1 to frame 30. (b) Optical flow of one frame. (c) Potential surface $\boldsymbol{E}$. (d) Potential surface $\boldsymbol{W}$. (e) The object-motion oriented map (OOM). (f) Quadtree partition on OOM. (g) The estimated inlier potential surface. (h) Inlier optical flow field. (i) The recovered local motion shown in 3D. (j) Outlier optical flow field. (k) Segmentation result.

We use this scene to evaluate the performance of our method in dealing with camera rotation. The segmentation results are shown in Fig. 8. From Fig. 8(a), we can see that the feature based methods place several checkerboards in the scene to obtain prominent feature points for motion segmentation. Our method, in contrast, does not rely on the strong features (corners and edges). From the original motion field Fig. 8(b), it is impossible to figure out what exact motions are involved in the scene. The OOM, however, has showed the foreground motions in Fig. 8(e) and the Quadtree detects them well in Fig. 8(f). From the estimated inlier potential surface Fig. 8(g) and its motion field Fig. 8(h), we can clearly find the rotation shape of the camera motion. This scene demonstrates the good performance of our method in dealing with camera rotation.

[3] 1T2TCRT-Parking sequence. This sequence is from the Complex Background dataset, which is utlized to evaluate the performance of our method in dealing with extremely challenging scenes with complex background and complex camera motions. Two kinds of motions are included in this scence, which are translation and rotation. Segmentation results shown in Fig.9 demonstrace that the OOM detects the moving objects in (e) even when the object is occluded and the background is complicated with complex camera motions. In addition, the inlier estimated in (h) is quite smooth and the two object motions are recovered precisely in (j). Segmentation result (k) demonstrates the effectiveness of our method in dealing with extremely challenging scenes.

6.3 Comparison with State-of-the-arts

In this subsection, our method is compared with several recent developed methods including: (1) joint inlier estimation and segmentation (GME-SEG) in [19], (2-3) iterative estimation based on least-square (LS) [16], and gradient decent (GD) [15], (4) outlier rejection filter (Filter) in [17], (5) RANSAC [18], and (6) the latest developed FOF [21], which is considered as the state-of-the-art in motion segmentation. In [21], two kinds of methods were developed: (1) FOF, which uses optical flow only, and (2) FOF+color+prior, which combines optical flow, color appearance and a prior model together. The source code of the first five methods can be found in [19]. The performances of FOF and FOF+color+prior presented in [21] are reported directly here. For the second type, our

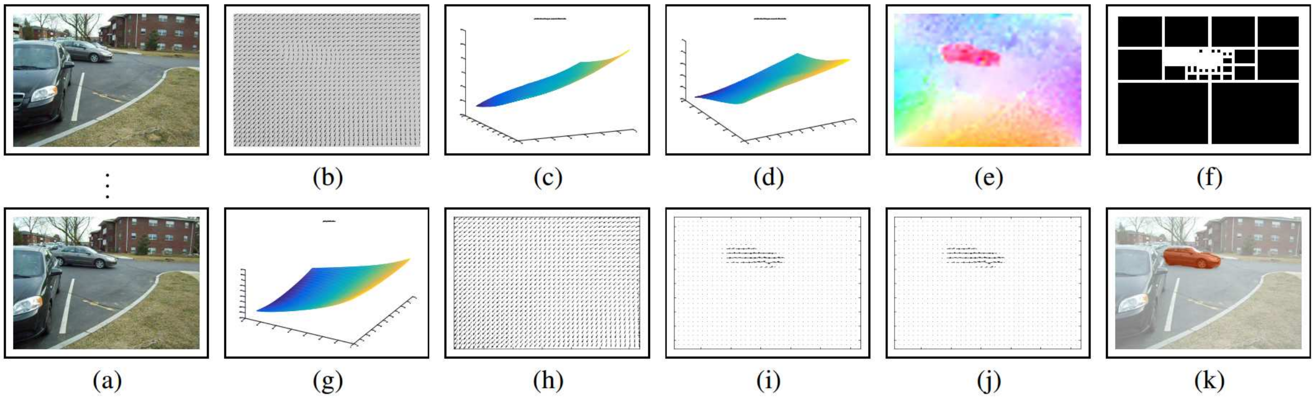


Fig. 9. Scenario 3: Parking sequence. (a) Image sequence from frame 1 to frame 30. (b) Optical flow of one frame. (c) Potential surface $\boldsymbol{E}$. (d) Potential surface $\boldsymbol{W}$. (e) The object-motion oriented map (OOM). (f) Quadtree partition on OOM. (g) The estimated inlier potential surface. (h) Inlier optical flow field. (i) The recovered local motion shown in 3D. (j) Outlier optical flow field. (k) Segmentation result.

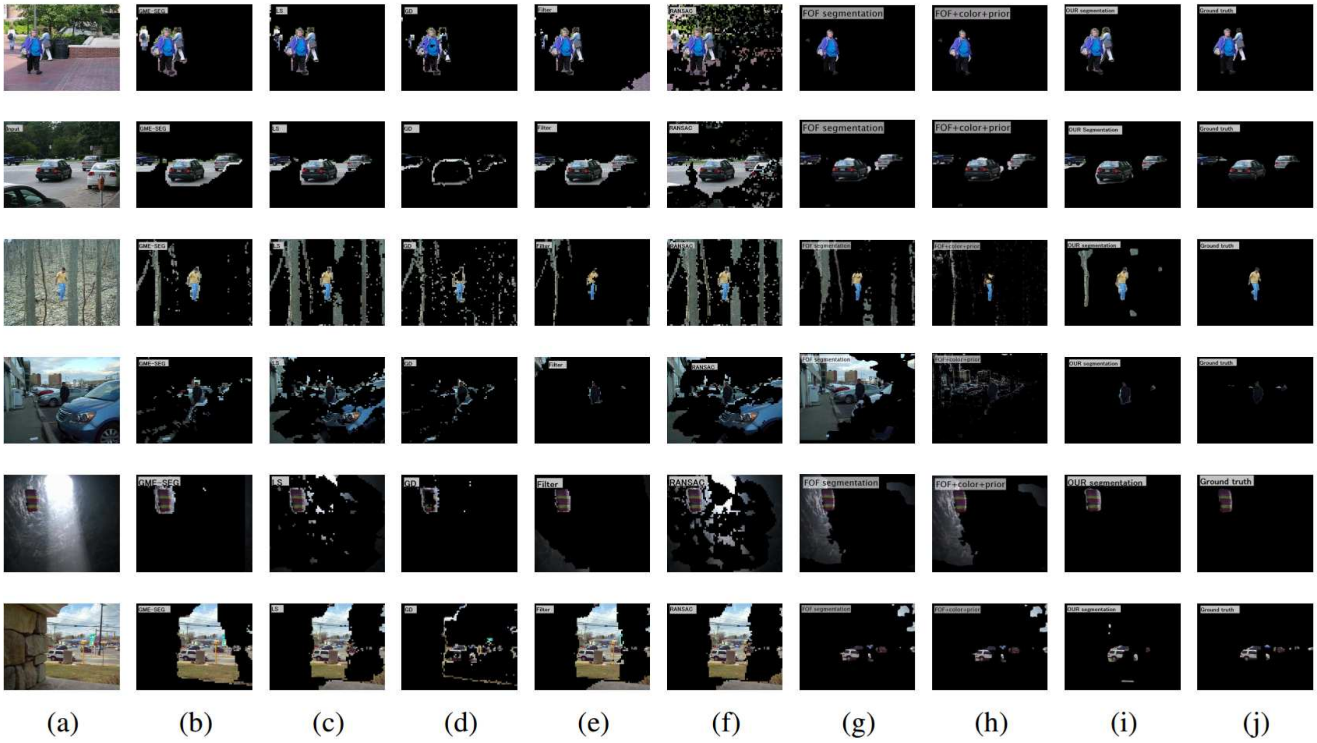


Fig. 10. Segmentation results of six existing dense based methods and ours on challenging scenarios. (a) Input sequences, from top to bottom: people2, cars2, forest, store, parachute, traffic. (b) Segmentation by GME-SEG [19]. (c) LS [16]. (d) GD [15]. (e) Filter [17]. (f) RANSAC [18]. (g) FOF [21]. (h) FOF+color+prior [21]. (i) Our segmentation. (j) Ground-truth segmentation.

method is compared with three well-known algorithms including: GPCA with spectral clustering [9], Local Subspace Affinity (LSA) [35] and RANSAC [18]. The source code of these methods is provided in [31].

The F-measure is utlized to evaluate the performance of each algorithm, which is calculated as following:

$$F = \frac{2 \times R_c \times P_r}{R_c + P_r}.$$

F-measure considers both the precision P_r and the recall R_c [36]. For dense-based methods, as segmentation is performed on each individual pixel, all pixels are utlized for F-measure calculation. While for feature based methods, only selected key feature points are involved in F-measure calculation. Table 1∼Table 3 report the F-measure of dense based methods on three benchmark datasets: Berkeley, Complex Background, and SegTrack, respectively.

From Table 1∼Table 3, it can be observed that our method achieves the highest performance for almost all videos, in which, 10% – 30% improvements on the sequences of cars 2, 3, 4, 7, and People 1 in the Berkely dataset, around 10% improvement on the sequences of drive, parking, and store in the Complex Background dataset, and more than 20% improvements on the sequences of parachutte and monkeydog in the SegTrack dataset. This result is quite appealing even when videos contain extremely challenging scenes, such as the ones with complex camera motions, complex backgrounds, occlusions. Quantitative results can be verified by the good visual quality of the segmentation results as shown in Fig. 10. Compared with the ground truth segmentation in Fig. 10(f), it can be found that our segmentation agrees with the true object regions more than existing methods.

With the help of the OOM and the Quadtree scheme, the accuracy of our algorithm is better than most existing methods. As shown in Figs.7–9, all the regions containing moving objects and depth discontinuities have been highlighted on OOM in (e) and labeled by Quadtree in (f). OOM separates inlier and outliers into homogeneous and heterogeneous motions, wich facilitates further segmentation. Besides, the Quadtree scheme based heterogeneous motion labeling method ensures the good performance of our method in estimation smooth and accurate camera-induced image motion.

From Table 1∼Table 3, we can also find that our method obtains poor performance on the cars 1, 9, 10, and girl sequences. The main reason is that the cars 1, 9 and 10 sequences contain some weak and smooth object motions, making inliers similar to outliers. For example, a big truck appears in cars 10 sequence, but the background is almost static is some frames. The motion field is still very smooth when these motions are mixed with inlier. They just appear in the curl-free and divergence-free components, but disappear in the OOM, which makes our method with poor performance. In addition, our method is performed based on the optical flow field, in which the performance is influenced by the accuracy of the optical flow estimation. Take girl sequence as an example, which captures a running girl in the sports yard. Since the girl is moving very fast, some frames taken by a moving camera are extremely fuzzy. The calculated optical flow fields become so noisy that no moving objects are identified. In this case, it is not enough to use optical flow information alonge. That is why FOF+color+prior, which utilizes additional information including color appearance and some prior models, performs better than ours. In addition, all methods perform poorly on these three videos (cheetah, penguin, monkeydog) of the SegTrack dataset, which is because these videos actually contain multiple moving objects, but only one primary object is labeled as the foreground in the ground truth.

7 Conclusions

In this paper, a prior-free dependent motion segmentation algorithm is proposed by introducing an HHD based OOM. For these three basic camera-induced image motions, HHD represents them in a uniform way without any prior-knowledge on camera motion and scene structure. The modified HHD identifies the camera-induced image motion (inlier) as one segment ir-

Table 1. F-measure of existing dense based methods and ours on Berkeley Motion Segmentation database

Sequences	GME-SEG	LS	GD	Filter	RANSAC	FOF	FOR+color	Our
Cars1	78.33	**86.18**	18.01	82.87	60.42	47.81	50.84	76.38
Cars2	55.90	65.97	15.70	78.28	34.21	46.37	56.60	**83.44**
Cars3	65.21	79.43	22.29	74.56	35.80	67.18	73.57	**87.60**
Cars4	45.69	49.78	22.96	77.22	22.81	38.51	47.96	**84.71**
Cars5	54.67	61.93	33.08	81.17	25.24	64.85	70.94	**85.10**
Cars6	33.01	51.23	28.77	57.53	13.40	78.09	84.34	**85.81**
Cars7	37.89	36.36	36.92	60.47	13.79	37.63	42.92	**86.50**
Cars8	62.20	81.24	8.57	78.44	37.02	87.13	87.61	**90.80**
Cars9	72.99	**80.99**	17.97	68.19	54.69	68.99	66.38	77.52
Cars10	60.01	66.04	14.34	**90.95**	81.78	53.98	50.84	54.93
People1	34.11	38.32	40.76	71.84	12.06	56.76	69.53	**80.14**
People2	78.16	84.45	69.30	81.70	37.53	85.35	88.40	**89.91**

Table 2. F-measure of existing dense based methods and ours on Complex Background database.

Sequences	GME-SEG	LS	GD	Filter	RANSAC	FOF	FOR+color	Our
drive	8.14	6.30	41.18	32.40	5.76	30.13	61.80	**83.33**
forest	15.42	11.01	15.41	19.87	10.67	19.48	31.44	**35.81**
parking	33.20	21.57	43.84	62.29	17.47	43.47	73.19	**83.57**
store	14.39	10.94	32.10	29.32	9.68	28.46	70.74	**80.50**
traffic	14.77	15.80	34.55	15.04	15.49	66.08	71.24	**71.77**

Table 3. F-measure of existing dense based methods and ours on SegTrack database.

Sequences	GME-SEG	LS	GD	Filter	RANSAC	FOF	FOR+color	Our
birdfall2	9.39	3.84	0.99	64.00	3.25	68.68	75.69	**76.23**
girl	22.51	20.26	15.36	18.21	12.33	75.73	**81.95**	78.53
parachute	23.01	18.97	12.88	16.30	44.03	51.49	54.36	**86.72**
cheetah	21.33	14.93	43.59	12.05	9.85	12.68	22.31	**55.77**
penguin	10.34	18.84	15.34	5.53	18.66	14.74	20.71	**21.90**
monkeydog	22.29	20.74	16.46	18.93	12.31	10.79	18.62	**45.84**

respective of depth variations, which ensures the effectiveness of our method in dealing with real-world scenes. The heterogeneous motion caused by moving objects and depth discontinuities cannot be represented by HHD, which will utlized for futher OOM construction. In the next, a data-driven Quadtree scheme is adopted to label the heterogeneous motion on OOM. After that, surface fitting based on a low-order polynomial function is employed for inlier estimation, which compensates depth discontinuities and separates them from true moving objects. Compared with existing work, our algorithm is prior-free and suitable for any kinds of camera motion under both 2D and 3D scenes, which be regarded as a general framework for dependent motion segmentation. Extensive experimental results demonstrated the effectiveness of our proposed method on robust segmentation compared with the state-of-the-arts.

References

[1] Cucchiara R, Prati A, Vezzani R. Real-time motion segmentation from moving cameras. *Real-Time Imaging*, 2004, 10(3):127–143.

[2] Li H, Wu W, Wu E. Robust interactive image segmentation via graph-based manifold ranking. *Computational Visual Media*, 2015, 1(3):183–195.

[3] Zhang F L, Wang J, Zhao H, Martin R R, Hu S M. Simultaneous camera path optimization and distraction removal for improving amateur video. *IEEE Transactions on Image Processing*, 2015, 24(12):5982.

[4] Zhang F L, Wu X, Zhang H T, Wang J, Hu S M. Robust background identification for dynamic video editing. *Acm Transactions on Graphics*, 2016, 35(6):197.

[5] Zhang Y, Tang Y L, Cheng K L. Efficient video cutout by paint selection. *Journal of Computer Science and Technology*, 2015, 30(3):467–477.

[6] Zografos V, Nordberg K. Fast and accurate motion segmentation using linear combination of views. In *Proc. of 22nd British Machine Vision Conference(BMVC)*, January 2011, pp. 12.1–12.11.

[7] Aldroubi A. A review of subspace segmentation: Problem, nonlinear approximations, and applications to motion segmentation. *ISRN Signal Process*, 2013, pp. 1–13.

[8] Ichimura N. Motion segmentation based on factorization method and discriminant criterion. In *Proc. of the Seventh IEEE International Conference on Computer Vision (ICCV)*, September 1999, pp. 600–605.

[9] Vidal R, Ma Y, Soatto S. Two-view multibody structure from motion. *International Journal of Computer Vision (IJCV)*, 2006, 68(1):7–25.

[10] Sugaya Y, Kanatani K. Geometric structure of degeneracy for multi-body motion segmentation. In *Proc. of 2nd International Workshop on Statistical Methods in Video Processing*, May 2004, pp. 13–25.

[11] Costeira J, Kanade T. A multibody factorization-method for independently moving objects. *International Journal of Computer Vision (IJCV)*, 1998, 29(3):159–179.

[12] Kanatani K, Matsunaga C. Estimating the number of independent motions for multibody motion segmentation. In *Proc. of the 5th Asian Conference on Computer Vision*, February 2002, pp. 7–12.

[13] Shi F, Zhou Z, Xiao J, Wu W. Robust trajectory clustering for motion segmentation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 3088–3095.

[14] Ochs P, Malik J, Brox T. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014, 36(6):1187–1200.

[15] Su Y, Sun M T, Hsu V. Global motion estimation from coarsely sampled motion vector field and the applications. *IEEE Transactions on Circuits Systtem and Video Technology*, 2005, 15(2):232–242.

[16] Smolic A, Hoeynck M, Ohm J R. Low-complexity global motion estimation from p-frame motion vectors for mpeg-7 applications. In *Proc. of International Conference on Image Processing (ICIP)*, September 2000, pp. 271–274.

[17] Chen Y M, Bajic I V. Motion vector outlier rejection cascade for global motion estimation. *IEEE Signal Processing Letters*, 2010, 17(2):197–200.

[18] Fischler M, Bolles R. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981, 26:381–395.

[19] Chen Y M, Bajic I V. A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field. *IEEE Transactions on Circuits Systtem and Video Technology*, 2011, 21(9):1316–1328.

[20] Qian C, Bajić I V. Global motion estimation under translation-zoom ambiguity. In *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, August 2013, pp. 46–51.

[21] Narayana M, Hanson A, Learned-Miller E. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 1577–1584.

[22] Tong Y, Lombeyda S, Hirani A N, Desbrun M. Discrete multiscale vector field decomposition. In *ACM transactions on graphics (TOG)*, volume 22, 2003, pp. 445–452.

[23] Polthier K, Preu E. Identifying vector fields singularities using a discrete Hodge decomposition. *In: Hege, H.C.,Polthier, K. (Eds.), Visualization and Mathematics III.*, 2003, pp. 123–134.

[24] Irani M, Anandan P. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, 20(6):577–589.

[25] Brox T, Bruhn A, Papenberg N, Weickert J. High accuracy optical flow estimation based on a theory for warping. In *Proc. of 8th European Conference on Computer Vision (ECCV)*, May 2004, pp. 25–36.

[26] Helmholtz H. über integrale der hydrodynamischen gleichungen, welche den wirbelbewegungen entsprechen. *Journal Für Die Reine Und Angewandte Mathematik*, 2010, 1858(55):25–55.

[27] Bhatia H, Norgard G, Pascucci V, Bremer P T. The helmholtz-hodge decomposition - a survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2013, 19(8):1386–1404.

[28] Baker S, Scharstein D, Lewis J P, Roth S, Black M J, Szeliskiv R. A database and evaluation methodology for optical flow. *Internations Journal of Computer Vision*, 2011, 92:1–31.

[29] Liang X, Zhang C, Matsuyama T. Inlier estimation for moving camera motion segmentation. In *Proc. of Asian Conference on Computer Vision (ACCV)*, November 2014, pp. 352–367.

[30] Liang X, Zhang C, Matsuyama T. A general inlier estimation for moving camera motion segmentation. *IPSJ Transactions on Computer Vision and Applications*, 2015, 7:163–174.

[31] Tron R, Vidal R. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007, pp. 1–8.

[32] Brox T, Malik J. Object segmentation by long term analysis of point trajectories. In *Proc. of 9th European Conference on Computer Vision (ECCV)*, September 2010, pp. 282–295.

[33] Tsai D, Flagg M, Rehg J. Motion coherent tracking with multi-label mrf optimization. In *Proc. of the British Machine Vision Conference*, January 2010, pp. 56.1–56.11.

[34] Liang X, McOwan P, Johnston A. A biologically inspired framework for spatial and spectral velocity estimations. *Journal of the Optical Society of America A*, 2011, 28(4):713–723.

[35] Yan J, Pollefeys M. A general framework for motion segmentation: Independent, articulated,

rigid, non-rigid, degenerate and non-degenerate. In *Proc. of 9th European conference on computer vision (ECCV)*, May 2006, pp. 94–106.

[36] Dembczynski K, Jachnik A, Kotlowski W, Waegeman W, Hullermeier E. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *Proc. of 30th International Conference on Machine Learning (ICML)*, June 2013, pp. 1130–1138.

Cuicui Zhang received the Ph.D degree in Computer Science from the Kyoto University, Japan, in 2015. She is currently an Assistant Professor in the School of Marine Science and Technology, Tianjin University. Her research interests are Computer Graphics and Visualization.

Zhilei Liu received the Ph.D. degree in Computer Science from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 2014. He is currently an Assistant Professor with the School of Computer Science and Technology, Tianjin University. His research interests cover multimedia computing, affective computing, machine learning, and pattern recognition.