Color Segmentation-based Optical Flow Computation and Motion Segmentation

Morimichi Nishigaki
Department of Computer Science
University of Maryland, College Park, MD 20742
michi@cs.umd.edu

Abstract

Finding corresponding regions in the frames of an image sequence is important for optical flow computation and motion segmentation. The problem is difficult because the shape and pattern of regions change over time by projective transformation and occlusions. In this paper, a segmentation algorithm which combines optical flow estimation with color segmentation is proposed. The projective transformation for each color segment between frames is estimated by iterative weighted least squares minimization and outlier elimination. Variable weight adjustment and outlier elimination are developed to exclude occlusion regions from the minimization process. Robustness of the proposed algorithm to occlusions is verified in experiments. The color segments are merged based on motion similarity, and dense optical flow is computed for the merged segments. Motion segmentation is performed by clustering of estimated transformations. Occlusions are identified from the estimated motion of the segments.

1 Introduction

Motion segmentation is an important pre-processing step for several applications in computer vision, such as tracking, action recognition, and compression of image sequences. The desired segmentation for these applications is a partitioning of the image into individual moving objects. Since the projective motion of a moving object is represented by parametric motion model, the

success of motion segmentation depends on the quality of estimation of the motion model parameters.

The task of computing motion in images is to find correspondences in the image sequence. The displacement of a point or a region in image is computed based on the similarity of image patterns. The dense field of displacement vectors is referred as optical flow. One problem in computing optical flow is ambiguity of correspondences. In general, a unique matching is not found because of the aperture problem. It means that local information is not enough to determine the displacement vector, such as in the motion of a line or in repeated patterns. Another problem is occlusions where correspondence do not exist. On the other hand, occlusions carry information about the structure of the scene, which is useful for motion segmentation [10]. In addition to these problems, since the size and shape of an object in an image is not consistent over time due to the projective transformation, simple two-dimensional template matching does not produce good optical flow.

Motion segmentation amounts to splitting the image into regions based on the motion in the image sequence. Usually, each segment indicates a region in which the objects' motion is represented by a common motion model. Since motion segmentation is based on image motion measurements, there same problem as the computing optical flow exists. Layer representation [13] is a popular concept in motion segmentation. The representation of each motion in the scene is referred as a layer. The motion segmentation task in the layer representation is to determine the layer descriptions and assign each pixel in the image sequence to the corresponding layer.

In this paper, an algorithm for computing a parametric motion field utilizing color segmentation technique is proposed, and dense optical flow estimation using parametric motion models is used for motion segmentation. The color segmentation divides images into regions based on the color homogeneity. The proposed algorithm integrates color cues with motion cues by utilizing the color segmentation for motion computation. It is assumed that each segment obtained by the color segmentation is a projection of a part of a plane. This assumption is not true in general, however, a small region in the image is approximately planar. By performing an over-segmentation at the color segmentation, the assumption most likely holds. Motion parameters for each segment are computed by segment-based matching taking the projective transformation and occlusions into account. This is achieved by iterative weighted least squares error minimization and outlier elimina-

tion that is designed to be robust to occlusions and initial estimates. Layer descriptions of motion segmentation are determined by clustering segments after color segments are merged based on similarity of motion. Occlusions are detected by finding overlaps of merged segments and isolated pixels.

2 Previous Work

2.1 Optical Flow

A brief overview of prior work on the optical flow problem is given here. Though exhaustive optical flow algorithms are reviewed in several review papers [2], [11]. There are two popular approaches to compute optical flow. One is the differential approach, and the other is the parametric approach.

Differential Approach

In the differential approach, optical flow is computed based on the image derivatives with respect to position and time. Assuming brightness preservation, the following constrain is derived by truncating second and higher order terms in the Taylor expansion of the differential of the image intensity function $I(x + dx, y + dy, t + dt)$ at (x, y, t).

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial x}v = -\frac{\partial I}{\partial t} \tag{1}$$

Here, I is the intensity function $I(x, y, t)$ at a point (x, y) at time t, and the vector $(u, v) = (dx/dt, dy/dt)$ is the pixel's displacement. This equation is referred to as brightness constancy constraints. Since the brightness constancy constrain is underdetermined, it is only possible to compute the flow vector that is perpendicular to the image edge, which is called the normal flow. In order to compute the optical flow vector (u, v), an additional constraint is required. The additional constraint is usually derived from assuming smoothness of motion over the image. For example, Lucas-Kanade algorithm [5] assumes that flow vectors are constant within a certain support window. Since this method computes the flow based on only information within the window, the ambiguity of motion due to the aperture problem is not solved. Instead of using local support, there is another approach where optical flow is computed by optimizing a global function which is incorporate with the brightness constancy constraint. Various methods in this category

optimize discontinuity preserving smoothness term as proposed by Horn and Schunck [8]. Nagel [7] also formulate the problem as a minimization of a global function with second order derivatives. A problem of the differential approach is the difficulty of computing accurate numerical differentiation because of noise, aliasing, or low frame rate.

Parametric Approach

In the parametric approach, image motion is represented by parametric motion models. Methods categorized as region-based matching [2], model the motion of a region in an image by its shift $d = (d_x, d_y)$ in the image. Black and Jepson [3] proposed a method using variable-order model fitting. They segment the image into regions of homogenous color and compute coarse optical flow. Motion model parameters are estimated for each individual color segment based on the coarse optical flow. A lower order motion model is first tried to fit coarse optical flow, then a higher order model is applied if the fitting error decreases. They use the following eight-parameter model, and use only $[a_0, a_3]$ or $[a_0, a_1, a_2, a_3, a_4, a_5]$ for the lower order model. These models correspond to translational and affine motion respectively.

$$\begin{aligned} u &= a_0 + a_1 x + a_2 y + a_6 x^2 + a_7 xy \\ v &= a_3 + a_4 x + a_5 y + a_6 xy + a_7 y^2 \end{aligned} \tag{2}$$

where the a_i are the parameters to describe motion of the region. (u, v) are the optical flow vectors at the image point (x, y). The eight-parameter motion models is derived by substituting the planar model $\alpha x + \beta y + \gamma = 1/Z$ into the infinitesimal projective motion formula [12].

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} t_z x - t_x \\ t_z y - t_y \end{pmatrix} + \begin{pmatrix} xy\omega_x - (x^2+1)\,\omega_y + y\omega_z \\ -xy\omega_y + (y^2+1)\,\omega_x - x\omega_z \end{pmatrix} \tag{3}$$

Here, α, β, and γ are parameters of a plane, $(u, v)^T$ is the motion vector on the image, $(t_x, t_y, t_z)^T$ is a translation vector, $(\omega_x, \omega_y, \omega_z)^T$ is a rotation vector to describe motion of the plane. $(x, y)^T$ is the location on the image. In the above equations, a normalized camera is assumed so that $x = X/Z$ and $y = Y/Z$ for the three-dimensional point (X, Y, Z) on the plane. Ji and Fermüller [9] show rank constraints on three-dimensional shape parameters of planar patches in multiple frames that are based on the above motion equation. Together with color segmentation, accurate three-dimensional motions of segments are estimated based on the constraint.

2.2 Motion Segmentation

Motion segmentation of images refers here to partitioning an image into regions of homogenous two-dimensional apparent motion. The basic assumption is that the homogenous apparent motion for each segment is represented by a different motion model. The motion models approximate the projective motion of the three-dimensional motion of objects.

Wang and Adelson [13] proposed a motion segmentation algorithm that represents moving images with sets of overlapping layers. They, first, compute optical flow, then estimate the affine motion parameters for every square non-overlapping region distributed over the image to fit the optical flow. After eliminating unreliable motion models for which the fitting error exceeds a certain threshold, the motion models are grouped in the affine motion parameter space by k-means clustering.

The algorithm proposed by Xiao and Shah [14] starts with tracking Harris corners to generate seed regions with affine motion models. Initial layers are obtained by expanding the seed regions to neighboring regions in which motion is represented by the same affine motion. Layers are merged based on the overlap of layers and number of pixels supporting the same affine motion. Then, multi-frame layer segmentation is obtained by graph cuts. The energy function for the graph cuts considers occlusion, and occlusions are identified by assigning an occlusion label.

Bleyer, Gelautz and Rhemann [4] proposed a color segmentation-based optical flow computation algorithm and applied it to motion segmentation. They track feature points and estimate affine motion parameter for each color segment by least squares error fitting of all correspondences found inside a segment. Then, layers are extracted by mean-shift clustering in eight-dimensional space that consists of six parameters of affine motion parameters and two parameters of the segment center.

Ogale, Fermüller and Aloimonos [10] classify motion segmentation problems into three categories based on motion direction and ordinal depth of background and object. They reveal the usefulness of occlusions in motion segmentation when the background and the object are moving in same direction. They proposed a motion segmentation algorithm to extract independently moving object regions using an ordinal depth conflict deduced by occlusion filling.

3 Segment-based Optical Flow Computation

In this paper, the following algorithm is proposed; the first stage of the algorithm is to segment the first of two consecutive images into regions of homogeneous color. Then, the parametric motion for each segment is estimated taking occlusions into account. Third, the segments of homogeneous motion are merged, and the motion parameters are recomputed. Then, dense optical flow is estimated within regions. The estimated parametric motion field is used for motion segmentation. The overview of the proposed algorithm is depicted in Figure 1.

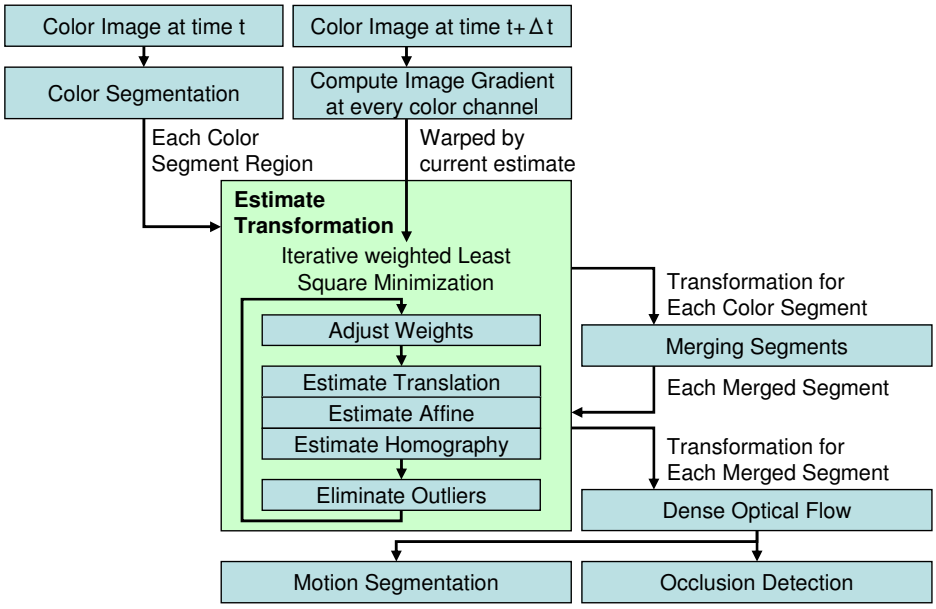


Figure 1: Overview of the proposed algorithm

3.1 Color Segmentation

It is assumed that the motion boundary coincides with the color boundary. This assumption is disputable. However, here the goal is restricted to computing apparent motion of image patterns. In this sense, the apparent motion of a rotating sphere in uniform color is zero. Since natural scenes usually have

enough structure, the optical flow field represents a good approximation of the real motion.

A color segmentation technique is incorporated into the algorithm. Given two consecutive images, the first image is segmented by color homogeneity. At the optical flow computation step, it is assumed that each segment is the projection of a part of a plane. Therefore, an over-segmentation of the image is preferable to ensure the assumption.

In principle, any color segmentation technique is applicable to the algorithm. The current implementation uses the graph-based segmentation algorithm proposed by Felzenszwalb and Huttenlocher [6]. The resulting color segmentation for the 'Mobile Calendar' image is shown in the Figure 2. The pixels drawn in the same color belong to the same segment in the figure.

(a) (b)

Figure 2: Color Segmentation. (a) The original image. (b) Computed color segmentation by the algorithm proposed by Felzenszwalb et al [6]

3.2 Segment-based Matching

Under the assumption that every segment is the projection of a part of a plane, the motion of a segment in the image sequence is represented by a homography. The motion of the segment is determined by finding the optimal transformation parameters by which the segmented region in the current frame is warped to the corresponding region in the previous frame. The optimal transformation parameters for each segment are defined as the parameters that minimize the following error between the segmented region

in the previous frame and the warped current frame to the previous frame:

$$E(m) = \sum_{p \in \mathcal{S}} \left[I_{t+\Delta t}(W(p;m)) - I_t(p) \right]^2 \quad (4)$$

Here, $\mathcal{S}$ is the set of pixels within the segment. $I_t(p)$ is the scalar intensity value of the image at pixel p at time t. This is extended to color vectors later. $W(p;m)$ denotes the warp represented by the transformation parameter vector m, which warps the point $p = (p_x, p_y)$ at time t to the corresponding point at time $t + \Delta t$. Since the error function is nonlinear, the optimal parameters that minimize the function are not solved analytically. The error is minimized iteratively by estimating incrementally parameter vector δm assuming the current estimate m is known starting with an initial parameter vector. The error function is rewritten with the incremental parameters as follow:

$$E(m + \delta m) = \sum_{p \in \mathcal{S}} \left[I_{t+\Delta t}(W(p;m+\delta m)) - I_t(p) \right]^2 \quad (5)$$

The Taylor expansion of $I_{t+\Delta t}(W(p;m+\delta m))$ is represented as follows:

$$\begin{aligned} & I_{t+\Delta t}(W(p;m+\delta m)) \\ = \; & I_{t+\Delta t}(W(p;m)) + \frac{\partial I_{t+\Delta t}(W(p;m))}{\partial m} \delta m + O\left(\delta m^2\right) \end{aligned} \quad (6)$$

The second term is the partial derivative of the image intensity function with respect to the transformation parameters. This can be rewritten using image gradient.

$$\frac{\partial I_{t+\Delta t}(W(p;m))}{\partial m} = \frac{\partial I_{t+\Delta t}(W(p;m))}{\partial p} \frac{\partial W(p;m)}{\partial m} \quad (7)$$

For simplicity, $\frac{\partial I_{t+\Delta t}(W(p;m))}{\partial p}$ is written as $\nabla I_{t+\Delta t}$ in the equations bellow. The term $\nabla I_{t+\Delta t}$ is the image gradient at the warped point $W(p;m)$ at time $t + \Delta t$. Assuming that the incremental parameters are enough small to ignore higher than second order terms in equation (6), the error function is approximated as follows:

$$\tilde{E}(m + \delta m) = \sum_{p \in \mathcal{S}} \left[I_{t+\Delta t}(W(p;m)) + \nabla I_{t+\Delta t} \frac{\partial W(p;m)}{\partial m} \delta m - I_t(p) \right]^2 \quad (8)$$

The incremental parameters that minimize the error function are solved by setting the partial derivatives of the approximate error function to zero.

$$\begin{aligned}
\frac{\partial \tilde{E}(m)}{\partial \delta m} &= 2\sum_{p\in\mathcal{S}}\left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right]^T \\
&\quad \left[I_{t+\Delta t}(W(p;m)) + \nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m} - I_t(p)\right] \qquad (9)\\
&= 2\sum_{p\in\mathcal{S}}\left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right]^T \left[I_{t+\Delta t}(W(p;m)) - I_t(p)\right] \\
&\quad + \left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right]^T \left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right]\delta m \\
&= 0 \qquad (10)
\end{aligned}$$

$$\delta m = H^{-1}\sum_{p\in\mathcal{S}}\left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right]^T [I_t(p) - I_{t+\Delta t}(W(p;m))] \qquad (11)$$

$$H = \sum_{p\in\mathcal{S}}\left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right]^T \left[\nabla I_{t+\Delta t}\frac{\partial W(p;m)}{\partial m}\right] \qquad (12)$$

Here, the H is an approximation of the Hessian of image intensity function $I_{t+\Delta t}(p,m)$. Then, the current estimate is updated as follows:

$$m \leftarrow m + \delta m \qquad (13)$$

The incremental minimization of the square error function derived above is known as Newton-Gauss method and has been discussed in the context of template pattern registration in images [5],[1]. The optimal transformation parameters represent the estimation of the motion of the segment in image sequence.

3.3 Robust Transformation Estimation

The method derived in the previous section is modified for better accuracy and robustness to the initial estimate of transformation parameters and occlusions.

Extension to Color Vectors

First, the error function in equation (4) is extended to deal with color cues. From point of view of least squares error minimization, p in the error function is a sampling point for model fitting. By sampling the intensity values in all three color channels, the error minimization becomes more reliable than using only gray level.

$$E(m) = \sum_{p \in \mathcal{S}} \sum_{c \in \{r,g,b\}} \left[I^c_{t+\Delta t}(W(p;m)) - I^c_t(p) \right]^2 \tag{14}$$

Here, $I^c_t(p)$ represents the intensity value of color channel c at point p at time t. The incremental transformation parameters for color image are derived by the same manner as described in the previous section.

$$\begin{aligned} \delta m &= H^{-1} \sum_{p \in \mathcal{S}} \sum_{c \in \{r,g,b\}} \left[\nabla I^c_{t+\Delta t} \frac{\partial W(p;m)}{\partial m} \right]^T \\ &\quad \left[I^c_t(p) - I^c_{t+\Delta t}(W(p;m)) \right] \end{aligned} \tag{15}$$

$$H = \sum_{p \in \mathcal{S}} \sum_{c \in \{r,g,b\}} \left[\nabla I^c_{t+\Delta t} \frac{\partial W(p;m)}{\partial m} \right]^T \left[\nabla I^c_{t+\Delta t} \frac{\partial W(p;m)}{\partial m} \right] \tag{16}$$

Robustness to Initial Parameters

Next a modification is made to overcome the sensitivity to the initial parameters. Since general projective motion of points on a plane is represented by a homography transformation, the dimension of the transformation parameters is eight. A simple experiment shown in Figure 3 reveals that this eight-dimensional minimization requires a good initial guess, less than two pixels in translation. In the experiment, the dark region in Figure 3(a) is the target segment for which the transformation parameters from the target to the corresponding regions in Figures 3(b)-(e) are estimated. The dark regions in Figures 3(b)-(e) are horizontally translated with respect to Figure 3(a) by one to four pixels respectively. The estimated homography transformation from Figure 3(a) to each of Figures 3(b)-(e) is represented by the green rectangle. For the correct transformation, the green rectangle would be aligned with the boundary of the dark region of Figures 3(b)-(e). The dark region is drawn with gradation to avoid ambiguity of the transformation. The iterative error minimization process is started with the identical transformation. The result shows that the estimation is unstable when the initial transformation

parameters are more than two pixels in translation from the true transformation. Even for less than two pixels in translation, the minimization is not able to reach the global minimum. This phenomenon is explained by the complexity of the eight-dimensional search, where the algorithm is easily trapped by a local minimum. On the other hand, when the transformation is restricted to a translational transformation, the stability of estimation is significantly improved, even though the translational transformation is a subset of homography transformation as shown in Figures 3(f)-(k). The images in Figures 3(f)-(k) are copies of the images in Figures 3(b)-(e) respectively. The optimization successfully reaches the global minimum in all cases in the experiment. It would be obvious that the two-dimensional search in transla-

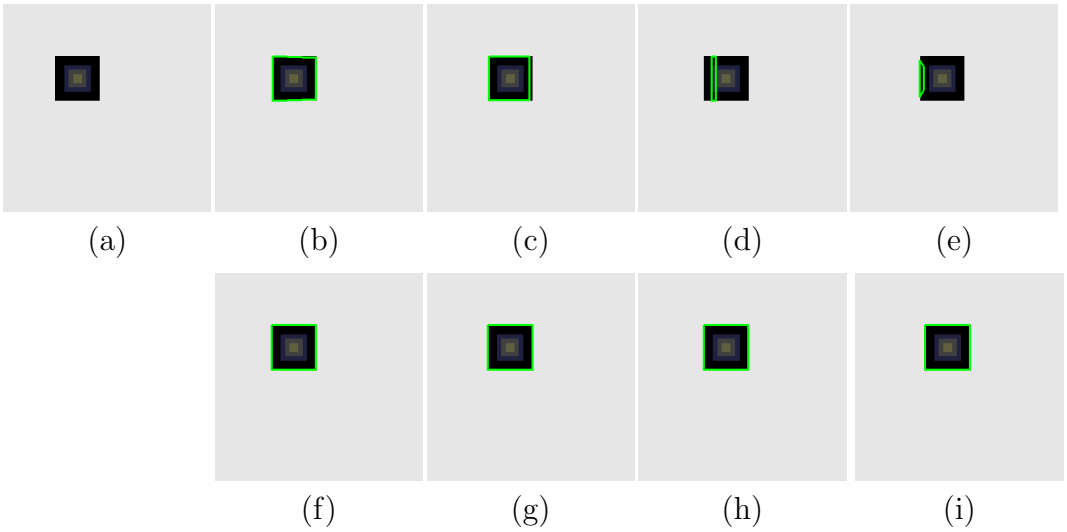


Figure 3: Demonstration of sensitivity to initial parameters. (a): The reference image. (b)-(e): The dark region is horizontally translated by one to four pixels with respect to the reference image, and the green distorted rectangle shows the segment boundary warped by the estimated homography transformation. (f)-(g): Same images as (b)-(e) but the estimation for the transformation is restricted to translation.

tion for the global minimum of the error function is more robust to the initial guess than an eight-dimensional search for the homography. However, the above simple experiments suggest a step by step minimization so that a good initial guess is obtained for the higher-dimensional search. The minimization

process is implemented in three steps. First, the translational transformation is estimated, and then using the estimated transformation as the initial parameters, an affine transformation is estimated. Finally using the estimated affine transformation, a homography transformation is estimated. Therefore, the function $W(p;m)$ is substituted by the following translational, affine and homography warp step by step.

$$W_t(p,m) = \begin{pmatrix} p_x + m_{13} \\ p_y + m_{23} \end{pmatrix} \tag{17}$$

$$W_a(p,m) = \begin{pmatrix} m_{11}p_x + m_{12}p_y + m_{13} \\ m_{21}p_x + m_{22}p_y + m_{23} \end{pmatrix} \tag{18}$$

$$W_h(p,m) = \frac{1}{m_{31}p_x + m_{32}p_y + 1} \begin{pmatrix} m_{11}p_x + m_{12}p_y + m_{13} \\ m_{21}p_x + m_{22}p_y + m_{23} \end{pmatrix} \tag{19}$$

The Jacobians that are required for the minimization in equation (15) and (16) are computed as follows:

$$\frac{\partial W_t(p,m)}{\partial m} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{20}$$

$$\frac{\partial W_a(p,m)}{\partial m} = \begin{pmatrix} p_x & 0 & p_y & 0 & 1 & 0 \\ 0 & p_x & 0 & p_y & 0 & 1 \end{pmatrix} \tag{21}$$

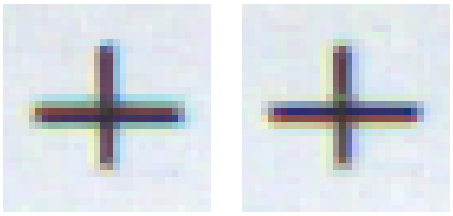
$$\frac{\partial W_h(p,m)}{\partial m} = \frac{1}{m_{31}p_x + m_{32}p_y + 1} \begin{pmatrix} p_x & 0 & -p_xq_x & p_y & 0 & -p_yq_x & 1 & 0 \\ 0 & p_x & -p_xq_y & 0 & p_y & -p_yq_y & 0 & 1 \end{pmatrix} \tag{22}$$

Here, $(q_x, q_y)^T = W_h(p;m)$.

Robustness to Outliers

Occlusions cause wrong estimation of the transformation because points are considered in the minimization that do not have matching points the previous frame. It is found that another problem in matching happens due to the color filter. A closer look at the images in Figure 4 shows that have different color patterns even though they are corresponding regions. The cross mark in the images is drawn in black on a white wall and captured in color images from different view points. Figures 4(a) and (b) shows different color patterns. This phenomenon is cased by the color filters by which the color on

a pixel is reconstructed from neighboring pixels. Due to the filtering, changes in the color pattern occur, mostly around the edges. The changes in color



(a) (b)

Figure 4: Color pattern changes due to color filter. (a),(b) color images of same scene, but different view points. A black cross mark is drawn on white wall in the scene.

pattern cause wrong estimation of the transformation. To overcome these issues, iterative outlier elimination is incorporated into the transformation estimation. Once the homography transformation that minimizes the error function is estimated, outliers in the error histogram are detected by assuming that the error caused by simple noise follows a normal distribution. The error, here, means each term before taking square in equation (14). That is the intensity difference at p, $I^c_{t+\Delta t}\left(W\left(p;m\right)\right) - I^c_t\left(p\right)$. Given the correct transformation, the difference is basically because of simple noise, and it is expected to be subject to a Gaussian distribution. However, the differences caused by other effects, such as occlusions or color filter, are not subject to the same distribution. In a typical error distribution in the experiments, about 2% of the samples are grater than three times the standard deviation. In contrast, the chance is about 0.3% if the noise is subject to a Gaussian distribution. This indicates that the error is not caused only by simple noise, but also by other effects. The causes of error are simply categorized into noise and other effects. Errors caused by noise are inliers, while errors caused by other effects are outliers in the estimation of the noise distribution. It is assumed that errors caused by noise are subject to a Gaussian distribution. Since most of the errors are caused by noise, and outliers lie on the tail of the distribution, the parameters of Gaussian distribution for the noise are estimated without the samples lying on its tails. The least median of squares fitting to the actual histogram of error is used because there is still the possibility that the outliers lie around the center of the distribution. The fitting

error for the i-th bin of the histogram is evaluated by the following value E_i.

$$E_i = N \int_{\mathcal{R}_i} \mathcal{N}(\tilde{\mu}, \tilde{\sigma}) - [\text{number of samples within } \mathcal{R}_i] \quad (23)$$

$$\mathcal{R}_i = [\mu + i\sigma/n, \mu + (i+1)\sigma/n] \quad (24)$$

Here, $\mathcal{N}(\tilde{\mu}, \tilde{\sigma})$ is the Gaussian distribution with mean $\tilde{\mu}$ and standard deviation $\tilde{\sigma}$. μ and σ are the mean and standard deviation for all samples, n is the number of bins of the error histogram. The equation means that the fitting error is evaluated as the difference between the expected number of samples within a range $\mathcal{R}_i$ and the actual number of samples within the range.

Since the chance that the absolute difference from the mean is grater than three times the standard deviation is less than 0.3%, it is assumed that such errors are caused by effects other than noise. Points that have such large error are excluded as outliers, and the minimization process to estimate the transformation is started over without those points.

$$\text{Outliers} = \left\{ p \;\middle|\; \left| I^c_{t+\Delta t} W(p; m) - I^c_t(p) - \mu^* \right| > 3\sigma^* \right\} \quad (25)$$

Here, μ^* and σ^* are the mean and standard deviation of the estimated Gaussian distribution of the noise that minimizes the median of squares of E_i. The estimation of the transformation and outlier elimination are iterated until the number of outliers or mean of squared error gets smaller than a certain threshold. The μ^* and σ^* are computed for each color channel only the first time for each segment, and the same parameters are used for later outlier elimination. Figure 5 is an example to show how the outlier elimination process works. Given a segment (for an example, see the green one in Figure 5(c)) obtained by color segmentation in the image of the Figure 5(a), the task here is to find the transformation of the segment from the segment in Figure 5(a) to the corresponding region in the image of Figure 5(b). The true transformation for the segment is identical transformation, however, a wrong transformation is computed by minimizing the error function because of occlusions at the right bottom of the segment, where the neighboring segment moves to upper left. The optical flow for points in the region is computed by the estimated transformation and shown in Figure 5(d). The flow vectors are color coded as denoted in Figure 5(g). Then, the intensity difference at each point on the segment at each color channel between the image of Figure 5(a) and the image of Figure 5(b) warped by the estimated transformation is computed. For each color channel, outliers

in the error are identified, and then points causing those extreme errors are excluded from the segment. The points on the segment after exclusion of the outliers are shown in Figure 5(e). Most of the occlusions are identified and excluded. A new transformation is estimated using only the remaining points. The resulting new transformation for the segment is shown in the Figure 5(f), which is exactly the same as the ground truth. In this example, no outliers are found any more in the second outlier elimination.

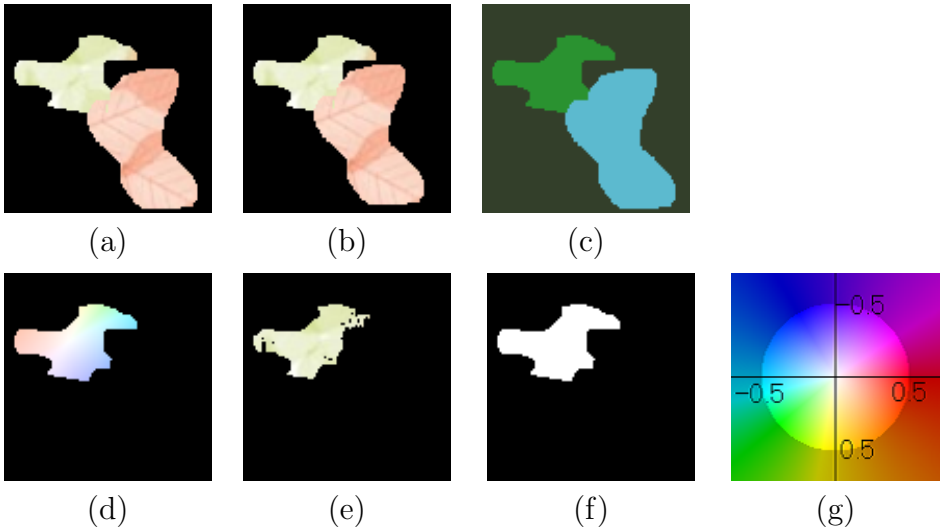


Figure 5: An example for outlier elimination. (a) A synthesized reference image. (b) A copy of the image (a), but the region with red texture is moved to upper left. (c) Color segmentation of the image (a). Points drawn in the same color belong to the same segment. (d) Optical flow from image (a) to (b) computed from the estimated transformation before outliers are eliminated. The optical flow vector on each point is color coded as shown in (g). (e) Pixels used to estimate transformation after outliers are eliminated. (f) Optical flow computed after outliers are eliminated. The white color codes zero flow.

The outlier detection works well when a relatively good transformation is estimated in the first iteration. In order to avoid the first estimation from being affected by occlusion, the error function in equation (14) is weighted and the weights are adjusted with less weights on points close to the boundary at the first time of iteration, because the occlusions are most likely close to

the boundary. Equation (14) is refined to

$$E(m) = \sum_{p \in \mathcal{S}} \sum_{c \in \{r,g,b\}} w(p) \left[I^c_{t+\Delta t}(W(p;m)) - I^c_t(p) \right]^2 \quad (26)$$

Here, $w(p)$ is the weight at point p. Two sets of weights are prepared in advance for the transformation estimation. Both of them assign lower weights at points close to boundary of the segment, but the one is smoother, and the other is steeper and the difference between the highest weight and the lowest weight is larger.

$$w(p) = \frac{1}{1 + \exp\{-c_1 (d(p)/d_{max} - c_2)\}} \quad (27)$$

Here, $d(p)$ is the distance from the boundary of the segment at a point p, and d_{max} is the maximum value of d over the segment. c_1 and c_2 are constant values. A lower value of c_1 is assigned for the smoother weight set, and a higher value is assigned for the steeper set. The steeper weight set is used first, and then a smoother weight set is used after outliers are eliminated.

3.4 Merging Segments

Segments which have similar transformation are merged. A target segment is merged with its neighboring segment if the transformation of the neighbor is similar to the target segment's transformation. Then, the region of the target segment is expanded to the neighboring segments that have similar transformation. Starting with the largest segment, the merging proceeds iteratively until no segments can be merged. When the segment l is the target segment, and the segment k is its neighbor, the similarity of transformations is measured by the following value $s(l,k)$.

$$s(l,k) = \frac{1}{|\mathcal{S}_k|} \sum_{(x,y) \in \mathcal{S}_k} |(u_l, v_l) - (u_k, v_k)| \quad (28)$$

$$w_l (u_l, v_l, 1)^T = H_l (x, y, 1)^T \quad (29)$$

$$w_k (u_k, v_k, 1)^T = H_k (x, y, 1)^T \quad (30)$$

Here, H_l and H_k are the matrix representation of the transformation for segment l and k respectively. $\mathcal{S}_k$ is the set of points on segment k, and $|\mathcal{S}_k|$ means the number of points on segment k. Merging occurs if the similarity value is smaller than a certain threshold. Then, the transformation of the merged segment is recomputed.

4 Motion Segmentation

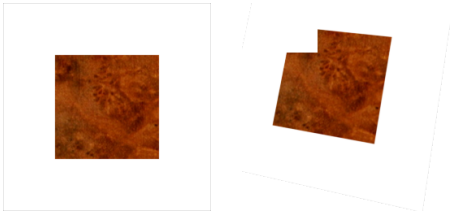
After the transformation has been estimated for each merged segment, the motion segmentation is computed by mean-shift clustering. The space for the clustering consists of the transformation parameters and the location of the segment. A cluster represents a layer in the motion segmentation.

Once the parametric motion from the previous frame to the current frame for each merged segment has been computed, pixels which belong to more than two segments in the current frame and pixels which do not belong to any segments are determined. Those pixels are labeled occlusions in the previous or current frame. The optical flow computed for pixels in occlusions is meaningless because no corresponding points exist for occlusions.

5 Experiments

The robustness of the proposed algorithm was verified by experiments using synthesized image sequences. A front-parallel plane of 2m width and 2m height with texture placed at 5m from the camera in the reference frame was simulated. The image in the reference frame is shown in Figure 6(a). Images of the plane at target frame were generated as translating and rotating the plane randomly. In the simulation, the rotation was restricted within the range of $-\pi/16$rad to $\pi/16$rad in each coordinate axis, and the translation was from -0.5m to 0.5m in each axis. The color segment corresponding to the plane was obtained in the reference frame. In order to simulate occlusions, another front parallel square plane without texture was placed in front of the textured plane in the target frame. The size of the non-textured plane was quarter of the textured plane in the reference frame. The non-textured plane centered at the upper-left corner of the textured plane in the target frame. 100 images of the plane with random translation and rotation at target frame were generated, and the transformations of the image of the plane from the reference frame to target frames were estimated by the proposed algorithm. The error of the estimated transformation was evaluated by the average distance between the four corner points of the estimated transformation and the ground truth in the target frame. In order to demonstrate the robustness clear, the same algorithm was applied to the same set of images without outlier elimination and weight adjustment. The statistics of the resulting error is shown in Table 1. 87% of trials successfully estimated the

transformation within 1 pixel accuracy. In contrast, 0% of trials had success with less than 1 pixel error without outlier elimination and weight adjustment. The result shows that the transformation estimation is significantly improved with the outlier elimination and weight adjustment.



(a) (b)

Figure 6: Verification for robustness. (a) The synthesized reference image of a textured plane. (b) An example image of the randomly translated and rotated plane occluded by a non-textured plane.

	mean [pixel]	median [pixel]	std. dev. [pixel]	cumulative distribution [%]				
				< 0.5	< 1.0	< 2.0	< 10.0	< 20.0
Propsed Algorithm	1.27	0.51	2.34	48	87	89	99	100
w/o outliers elimination	13.37	13.37	1.34	0	0	0	0	100

Table 1: Verification for robustness. Statistics of average error between true displacement and the estimated displacement at the four corner points of the plane in the 100 trials. The proposed algorithm is used to estimate displacements, but outlier elimination and weight adjustment are turned off in the second row.

The proposed algorithm was applied to real image sequence. Figures 7(a) and (b) show two consecutive frames of the image sequence, where the train and the ball move left, and the calendar moves down as the camera moves left backward. Figure 7(d) shows dense optical flow computed by the estimated transformation for each color segment shown in Figure 2(b). Flow vectors are color coded as denoted in Figure 7(c) The initial color segments were merged based on similarity of the transformation. Transformations for each

merged segments were estimated, and optical flow computed by the transformation is shown in Figure 7(e). Although the true motion is not known in this sequence, the computed dense optical flow seems to be approximating the scene motion very well. The motion segmentation was computed by mean-shift clustering. In this experiment, elements $(1, 3)$ and $(2, 3)$ of the homography matrix and the center of gravity of each segment were used for clustering. The resulting layers are shown in Figure 7(f). Pixels drawn in the same color belong to the same layer in the figure. Occlusions were detected by finding points where more than two merged segments overlapped or no segment covered. Occlusions are shown in Figure 7(f) in red. Since the camera was moving left backward in the sequence, the boundary of the image except right boundary was also detected as occlusion.

6 Conclusion

Utilizing a color segmentation technique, optical flow was computed by estimating the transformation of color segments between frames. In order to find region-based correspondences, a robust transformation estimation algorithm was proposed. The algorithm was based on the Newton-Gauss least squares minimization, and extended to be accurate and robust to occlusions and initial estimates. A variable weighting in the minimization and outlier elimination were developed to avoid wrong estimation due to occlusions. Significant improvement in robustness to occlusions was shown in experiments. The proposed algorithm was demonstrated on a real image sequence and dense optical flow in high quality has been computed. Motion segmentation and occlusion detection were performed based on the estimated parametric motion.

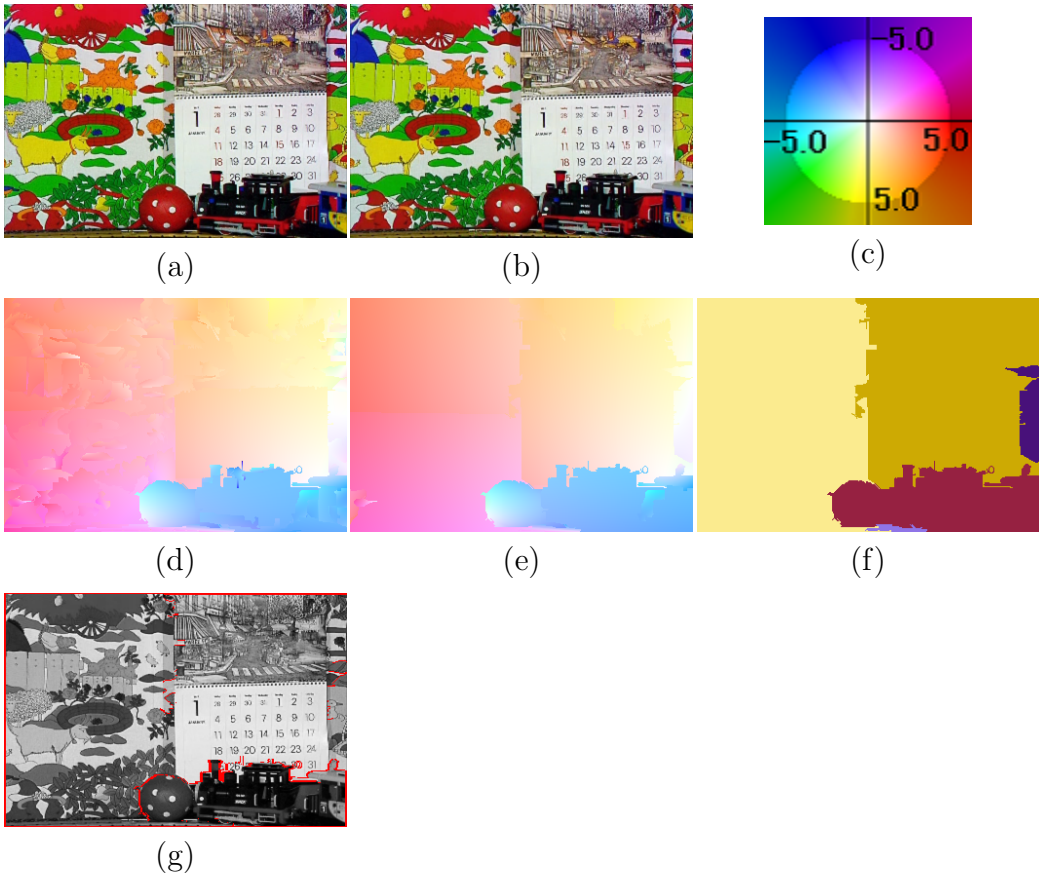


Figure 7: Real image experiment. (a,b) Two consecutive frames of image sequence, where the train and the ball move left, and the calendar moves down as the camera moves left backward. (c) The color code used in (d) and (e) to show optical flow. (d) The dense optical flow computed by the estimated transformation for each color segment. (e) The dense optical flow computed for each merged segment. (f) The motion segmentation. Pixels of the same color belong to the same layer. (g) Occlusions deduced from motions of merged segments. Pixels in red are detected as occlusions.

References

[1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2002.

[2] Fleet D.J. Barron, J.L. and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[3] Michael J. Black and Allan D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, 1996.

[4] Michael Bleyer, Margrit Gelautz, and Christoph Rhemann. Colour segmentation-based computation of dense optical flow with application to video object segmentation. *ÖGAI-Journal*, 24(1):11–15, 2005.

[5] Lucas B D and Kanade T. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.

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

[7] Nagel H. H. Displacement vectors derived from second-order intensity variations in image sequences. *Comput. Vision, Graphics, Image Processing*, 21(1):85–117, 1983.

[8] Berthold K.P. Horn and Brian G. Rhunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[9] Hui Ji and Cornelia Fermüller. A 3d shape constraint on video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):1018–1023, 2006.

[10] Abhijit S. Ogale, Cornelia Fermüller, and Yiannis. Aloimonos. Motion segmentation using occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6), June 2005.

[11] J.P. Lewis Stefan Roth Michael J. Black Simon Baker, Daniel Scharstein and Richard Szeliski. A database and evaluation methodology for optical flow. *In Proc. of the IEEE International Conference on Computer Vision*, October 2007.

[12] E. Truco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[13] J. Y. A. Wang and E. H. Adelson. Representing Moving Images with Layers. *The IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[14] Jiangjian Xiao and Mubarak Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1644–1659, 2005.