Motion Segmentation[1]

Cornelia Fermüller, Tomáš Brodský and Yiannis Aloimonos

Computer Vision Laboratory

Center for Automation Research

University of Maryland

College Park, MD 20742-3275

December 22, 1998

Abstract

Since estimation of camera motion requires knowledge of independent motion, and moving object detection and localization requires knowledge about the camera motion, the two problems of motion estimation and segmentation need to be solved together in a synergistic manner. This paper provides an approach to treating both these problems at the same time. The technique introduced here is based on a novel concept, "scene smoothness," which parameterizes the variation in estimated scene depth with the error in the underlying 3D motion. The idea is that incorrect 3D motion estimates cause distortions in the estimated depth map, and as a result smooth scene patches are computed as unsmooth, i.e., rugged, surfaces. The correct 3D motion can be distinguished, as it does not cause any distortion and thus gives rise to the smoothest background patches between depth discontinuities, with the locations corresponding to independent motion being unsmooth. The current implementation employs a binocular observer whose nature is exploited in the extraction of depth discontinuities, a step that facilitates the overall procedure, but the technique can be extended to a monocular observer in a variety of ways.

1 Introduction

The detection of independently moving objects has attracted a lot of attention because of its importance in visual processing and its potential for applications. This has led to the development of a number of algorithms, some of which show good performance under restricted conditions, but a general solution still remains elusive. The reason is that the problem is very difficult and the basic computational constraints underlying the problem have not yet been studied systematically. We give a synopsis of past approaches to independently moving object detection and discuss the various difficulties connected with the problem. For ease of discussion we shall use the term "moving object" to refer to any object moving with respect to stationary portions of the scene, which we refer to as the "background."

For a stationary observer, a simple approach is to difference images obtained over a short time span and mark the nonzero regions of the result. For a moving observer the problem is considerably more complex. The computation of egomotion parameters and the detection of moving objects are now interrelated. In order to detect moving objects well, egomotion estimates seem to be required, but a desirable condition for computing such estimates is the knowledge of where the moving objects are, due to the sensitivity of the egomotion estimation process. This makes it clear that a general, useful approach has to address the two processes simultaneously. What is needed is a study of the constraints arising from simultaneous estimation of egomotion and moving object detection.

Past approaches can be classified in two groups: (a) The first group of approaches bases the detection of moving objects, prior to 3D motion estimation, on 2D motion field measurements only [2–8]. A moving object generally gives rise to a discontinuity in the motion field along its boundary, which is used to signal the presence of a moving object. This approach has various difficulties. **First**, the detection of motion flow discontinuities without any additional knowledge about the 3D motion or scene is very difficult, as it requires the full motion flow to be computed, which in turn requires the integration of measurements (smoothing) over a local area [9, 10]. However, without prior knowledge of the motion boundaries the integration can easily lead to erroneous flow estimates exactly at the motion flow discontinuities. The estimation of motion flow can be improved [3, 11] by using robust statistical methods, but at the cost of intensive computation. Despite the variety of segmentation techniques being used, such as voting, clustering, Hough transform, Markov Random Field incorporating line processes, as well as using other cues such as intensity or color edges, detection performance is limited and often depends on the scene type. **Second**, both moving objects and depth discontinuities give rise to motion field discontinuities. Therefore, spurious segmentations are produced at depth discontinuities when the observer is translating. Due to the primitive measurements used, these methods cannot cope with such confusion.

(b) The second group of approaches assumes that partial or full information about egomotion is available or can be recovered [12–20]. Most of these studies assume the availability of a full motion field; for example, Thompson and Pong suggest in a pioneering study that if the egomotion parameters are known completely then the flow field can be checked for inconsistencies with the epipolar constraint arising from the presence of moving objects. Such approaches also run into problems at motion boundaries, where, as mentioned above, it is difficult to derive correct flow estimates. As the scene in view can take on any positive value, only the direction of the translational flow component, but not its length, is specified. Thus, at locations where the flow is not known correctly, the constraints suggested often reduce to those used in the image-based segmentation approach. Very seldom are normal flow measurements used as a basis for detection of moving objects. Considering only pointwise normal flow, the only computational constraint that can be imposed arises from the fact that the scene in view has positive depth which constrains the direction of the translational component of the normal flow to a half-plane. However, for many types of independent motion, a detection scheme based solely on this constraint may miss the moving object. If under some circumstances one can put a bound on how close or how far objects may be to the observer, then more constraints can be imposed by checking the estimates against these bounds [16].

It is clear that if the egomotion is known, or assumed to be computed accurately, the detection of moving objects and estimation of their motions becomes much easier; however, perceptual systems do not treat the problem in this way. A very interesting question is how perceptual systems are able to deal with both egomotion estimation and moving object detection.

A number of studies have proposed solutions that address these problems, but they are all based on either models about the scene in view or simplified models of image formation [21].

Such approaches have been used recently in integrated systems performing motion segmentation, saccades and pursuit [22–24]. The additional constraints imposed on the scene in view or image formation are justified only in restricted domains, such as applications dealing with aerial images. Since in such situations the variation in depth of the scene is small compared to the average depth of the scene, the motion of the background can often be compensated for by a registration process [5, 25–27]. Then any uncompensated regions correspond to moving objects.

There also have been several works proposing that improvement in motion detection can be gained if temporal integration is considered. However, the integration is usually done at the image processing level, using techniques from the target tracking and surveillance community or extending regularization to space-time [4, 6, 7, 25, 28]. Questions related to an integration via 3D motion and scene structure have not been sufficiently explored yet, as the extension of the rigidity constraint to multiple frames is non-trivial.

In this study, we propose an algorithm for motion segmentation that simultaneously addresses the problems of 3D motion estimation and detection of moving objects from image sequences without making any assumptions about the scene in view. The technique is developed for a binocular observer but the same ideas extend to the harder case of a monocular observer.

2 Motivation, Preliminaries and the Key Idea

2.1 What input

One may argue, as in several prior works, that motion segmentation arises naturally after an accurate description of the image motion field is obtained. It was argued in the previous section that accurate image flow is prevented by the lack of knowledge of the exact depth discontinuity locations. There exists however an additional reason that only recently was understood [29]. It has to do with the statistical difficulty of integrating local, one-dimensional motion signals into two-dimensional image velocity measurements. Any procedure for estimating image motion has to start with normal flow measurements, that is, the image motion component perpendicular to local edges. It has been shown [29] that when these local measurements are combined in a neighborhood to produce image motion, an estimate of flow is obtained which is biased. The estimated value depends on the distribution of image gradients, the actual flow and the error in the normal flow. This is strikingly observed in the Ouchi illusion (Figure 1). The pattern in Figure 1 has the surprising property that small motions can cause illusory relative motion between the inset and background regions [30]. The reason for this illusion is that for the particular spatial gradient distributions of the Ouchi pattern, the bias in the estimation of flow is highly pronounced, giving rise to a large difference in the velocity estimates in the two regions. Since the texture at an image patch influences the accuracy of image flow there—by contributing to a large or small bias—it is important that a motion segmentation algorithm rely—at least in its early stages—as little as possible on accurate flow values. Equivalently, it is worth addressing the problem starting with the normal flow values and whenever flow values at an image patch are utilized they should be accompanied by some measure of error indicating how well the obtained flow fits the data, since such an error contains valuable information.

[Figure 1 about here.]

2.2 The key idea

The solution procedure introduced in this paper solves for camera motion and independent motion at the same time. In so doing, it introduces a novel way of recovering 3D motion, which naturally handles detection of depth discontinuities and independent motion. Traditional ways of recovering egomotion are problematic in this case. The next paragraph explains why this is so, after introducing some symbolism. Then the new idea of the paper is described.

Consider a standard calibrated pinhole camera with focal length f, and the coordinate system $OXYZ$ attached to the camera with Z being the optical axis. Consider that the camera is moving with instantaneous translation $\mathbf{t} = (U, V, W)$ and rotation $\boldsymbol{\omega} = (\alpha, \beta, \gamma)$ measured in the system $OXYZ$ (Figure 2). Image points are represented as vectors $\mathbf{r} = [x, y, f]^{\mathrm{T}}$ where x, y the image coordinates of the point in the image Cartesian system ox, oy with $ox \| OX$, $oy \| OY$ and o the intersection of the Z axis with the image plane. A scene point $\mathbf{R}$ is projected onto image point $\mathbf{r} = \frac{f\mathbf{R}}{\mathbf{R}\cdot\hat{\mathbf{z}}}$ with $\hat{\mathbf{z}}$ the unit vector along the Z axis. As the camera moves, a scene point $\mathbf{R}$ moves with velocity $\dot{\mathbf{R}} = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{R}$. Then the image motion field is [31]:

$$\dot{\mathbf{r}} = -\frac{1}{(\mathbf{R}\cdot\hat{\mathbf{z}})}(\hat{\mathbf{z}} \times (\mathbf{t} \times \mathbf{r})) + \frac{1}{f}\hat{\mathbf{z}} \times (\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})) = \frac{1}{Z}\mathbf{u}_{\mathrm{tr}}(\mathbf{t}) + \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega}) \tag{1}$$

where Z the scene depth $(\mathbf{R}\cdot\hat{\mathbf{z}})$ and $\mathbf{u}_{\mathrm{tr}}, \mathbf{u}_{\mathrm{rot}}$ the direction of the translational and rotational flow, respectively [32]. The normal flow u_n at a point with image gradient $\mathbf{n}$ ($\mathbf{n}$ is a unit vector) is simply

$$u_n = \dot{\mathbf{r}} \cdot \mathbf{n} \tag{2}$$

In non-vector notation, (1) becomes:

$$\dot{\mathbf{r}} = \vec{u} = (u, v) = \frac{\mathbf{u}_{\mathrm{tr}}}{Z} + \mathbf{u}_{\mathrm{rot}} = \frac{W(\vec{p} - \vec{p_0})}{Z} + (\alpha\frac{xy}{f} - \beta(\frac{x^2}{f} + f) + \gamma y, \alpha(\frac{y^2}{f} + f) - \beta\frac{xy}{f} - \gamma x) \tag{3}$$

where $\vec{p_0} = (x_0, y_0) = (\frac{fU}{W}, \frac{fV}{W})$ the direction of translation and $\vec{p} = (x, y)$ the image point.

[Figure 2 about here.]

Traditionally, 3D motion is estimated by taking advantage of the epipolar constraint, which relates the image motion field to 3D motion. It is obtained by eliminating the depth of the scene from equations (1) and amounts to:

$$(\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}}(\mathbf{t})) \cdot (\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega})) = 0 \tag{4}$$

Usually, the distance of the flow vector $\dot{\mathbf{r}}$ from the epipolar line (determined by $\mathbf{u}_{\mathrm{tr}}(\mathbf{t})$ and $\mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega})$) is computed and the sum of the squared distances

$$\sum((\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}}(\mathbf{t})) \cdot (\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega})))^2 \tag{5}$$

is minimized. In practice, because minimization of (5) suffers from bias [33] a scaled version of (5) is used to provide an unbiased solution. Thus, estimates of the (scaled) translation $\hat{\mathbf{t}}$ and rotation $\hat{\boldsymbol{\omega}}$ are obtained from the minimization of:

$$E = \sum \frac{((\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})) \cdot (\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})))^2}{\|\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})\|^2} \tag{6}$$

Minimization of (6) provides reasonable results for 3D motion given a large field of view and a stationary scene, although often the confusion between rotation and translation may create problems.

In the case where the scene contains independent motion one may consider applying epipolar minimization to image patches. Consider dividing the image into a number of patches. Minimization of (6) in each patch could provide an answer for 3D motion. Then, presumably through an aggregation or clustering technique (such as Hough transform, for example) patches providing similar answers would constitute the background. Such an approach, however, is bound to fail in practice. It is by now common knowledge that the small amount of data in each patch will create confusion between the parameters of the 3D motion giving rise to unpredictable results. Besides, the quality of flow values in each patch, being at the mercy of the unknown location of depth boundaries and the image texture, can only worsen the situation.

[Figure 3 about here.]

There is, however, a new way of looking at this problem and that is among the main contributions of this paper. This new approach is achieved by involving the scene in view in the analysis of acquiring 3D motion on the basis of image motion [34].

Consider an estimate of the 3D rigid motion $\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}$. At an image point $\mathbf{r}$ where the normal flow direction is $\mathbf{n}$, the depth $\hat{Z}$ can be estimated from (2) as:

$$\frac{1}{\hat{Z}} = \frac{\dot{\mathbf{r}} \cdot \mathbf{n} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}}{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}} \tag{7}$$

where $\mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}), \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ the estimated rotational and translational flow, respectively.

Substituting the value of $\dot{\mathbf{r}}$ from (1) into (7), one obtains:

$$\frac{1}{\hat{Z}} = \frac{1}{Z} \frac{\mathbf{u}_{\mathrm{tr}}(\mathbf{t}) \cdot \mathbf{n} - Z \mathbf{u}_{\mathrm{rot}}(\delta\boldsymbol{\omega}) \cdot \mathbf{n}}{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}}$$

where $\mathbf{u}_{\mathrm{rot}}(\delta\boldsymbol{\omega})$ the rotational flow due to the rotational error $\delta\boldsymbol{\omega} = (\hat{\boldsymbol{\omega}} - \boldsymbol{\omega})$. The above equation can be further expressed as:

$$\hat{Z} = Z \cdot D \quad \text{with} \tag{8}$$

$$D = \frac{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{(\mathbf{u}_{\mathrm{tr}}(\mathbf{t}) - Z \mathbf{u}_{\mathrm{rot}}(\delta\boldsymbol{\omega})) \cdot \mathbf{n}} \tag{9}$$

where D hereafter is termed the distortion factor [35]. Equation (8) shows how wrong depth estimates are produced due to inaccurate 3D motion values. The distortion factor for any direction

$\mathbf{n}$ corresponds to the ratio of the projections of the two vectors $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\mathrm{tr}}(\mathbf{t}) - Z\mathbf{u}_{\mathrm{rot}}(\delta\boldsymbol{\omega})$ on $\mathbf{n}$. The larger the angle between these two vectors is, the more the distortion will be spread out over the different directions. Thus, considering a patch of a smooth surface in space and assuming that normal flow measurements are taken along many directions, a rugged (i.e., unsmooth) surface will be computed on the basis of wrong 3D motion estimates. Consider Figure 3. It shows one frame from a video sequence. For two different 3D motions, the depth $\hat{Z}$ is estimated and plotted on the corresponding image pixel. Notice the reasonably smooth estimates (separated by scene discontinuities) for the correct motion (Figure 3b) and compare with the sharp changes in the depth map in Figure 3c where an incorrect 3D motion is used.

This observation constitutes the main idea behind our approach. It amounts to searching for the 3D motion that will minimize some measure of depth variability. Consider Figure 4, showing an image divided into small patches. Consider further a search procedure that searches for the 3D motion which when used with the normal flow data inside the patch provides, from equation (1), depth values in the particular patch that vary the least. The 3D motion that minimizes depth variation in the largest possible number of patches constitutes the solution. This is easy to understand if the scene in view is smooth (containing no depth discontinuities) and static (no independent motion). If the scene in view contains depth discontinuities and is static, then there will be patches, namely the ones containing discontinuities, for which the correct 3D motion produces large variability of depth; but these patches can still be subdivided into two smaller patches, each on different sides of the discontinuity, for which small variability of depth can be achieved. This, however, in general cannot be accomplished if the patch contains independent motion.

[Figure 4 about here.]

The preceding discussion provides an intuitive understanding of the proposed solution. The image is subdivided into small patches and search for the camera motion is performed. For each candidate 3D motion, the depth values inside a patch are estimated and a measure of their variability is computed. Patches with high variability of depth are further subdivided until the smaller patches acquire small depth variation or until it becomes clear that depth variation cannot be decreased. This subdivision is facilitated by other processes that will be explained later. The correct 3D motion is the one that minimizes depth variation within the patches or their subdivisions. Patches corresponding to small depth variation constitute the background while patches that are unsmooth signify independent motion.

There are still two issues that need to be discussed before proceeding with the algorithm description and its analysis. These are (a) how is the search for camera motion going to be performed and (b) for which technical reasons minimization of depth variation provides the correct solution, and how is it related to the epipolar constraint? These are discussed in the next two sections. Then, a description of the algorithm is given and the rest of the paper is devoted to explaining in detail the algorithm's modules. The paper finishes with a number of experimental results.

3 Searching for the Camera Motion

This is the main part of the algorithm. Finding the camera motion by maximizing smoothness will simultaneously provide the background and the independently moving parts of the image. Camera motion consists of five parameters, two for the translation (FOE or FOC) and three for the rotation. It would be impractical to perform a search in five dimensions. Thus we search for the two parameters of the translation and for each candidate translation we find from the data the best corresponding rotation. Searching for the FOE amounts to searching for a point on the Gaussian sphere (hemisphere) and can be quite efficient through an appropriate parameterization. In our implementation, for the purpose of showing intermediate results of the algorithm in the form of retinotopic maps [36], we perform the search on an area of the plane which is four times as large as the image and centered at the image center.

Given a candidate translation, the rotation is computed with a detranslation technique introduced in [37]. This is achieved by examining normal flow vectors that are perpendicular to lines passing from the candidate FOE, since these values would not contain a translational part. Let point $\hat{\mathbf{t}}$ be a candidate FOE and consider the pencil of lines centered at $\hat{\mathbf{t}}$. Vectors perpendicular to such lines at point $\mathbf{r}$ are $\mathbf{v}(\mathbf{r}) = \hat{\mathbf{z}} \times \mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) = \hat{\mathbf{z}} \times (\hat{\mathbf{z}} \times (\hat{\mathbf{t}} \times \mathbf{r}))$. If the camera motion is $(\mathbf{t}, \boldsymbol{\omega})$, by projecting the flow $\dot{\mathbf{r}}$ onto $\mathbf{v}$ one obtains:

$$\dot{\mathbf{r}} \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|}(\frac{1}{Z} f(\mathbf{t} \times \hat{\mathbf{t}}) \cdot \mathbf{r} + (\boldsymbol{\omega} \times \mathbf{r}) \cdot (\hat{\mathbf{t}} \times \mathbf{r}))$$

By letting $\mathbf{t} = \hat{\mathbf{t}}$, the translational component of the projection of $\dot{\mathbf{r}}$ onto $\mathbf{v}$ becomes zero and the above equation simplifies to:

$$\dot{\mathbf{r}} \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|}(\boldsymbol{\omega} \times \mathbf{r}) \cdot (\hat{\mathbf{t}} \times \mathbf{r}))$$

This equation serves as a basis for estimating the rotation. At points with suitable normal flow directions we have equations

$$\dot{\mathbf{r}} \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|} = (\frac{1}{\|\mathbf{v}\|}(\mathbf{t} \times \mathbf{r})^{\mathrm{T}})\boldsymbol{\omega}$$

that are linear in $\boldsymbol{\omega}$. Thus, 3D motion can be estimated by fitting for each candidate translation the best rotation and checking the size of the residual fit.

The technique described above for finding the rotation makes sense only if the normal flow values used in the detranslation technique belong to the background area. As the final goal of the computation is to localize the background, this problem is especially difficult. Luckily, however, we do not need a very large amount of data to apply the detranslation technique. Solution to this problem comes from employing a binocular observer. Since a stereo pair is available, simple techniques can provide a number of major depth discontinuities. Without a doubt, very sophisticated and usually slow stereo algorithms could be employed in order to detect a large number of depth boundaries, but our interest here is to localize a part of the background in order to estimate the rotational parameters of the camera motion. Thus, a simple approach using horizontal disparities delivers some depth boundaries which also contain moving object boundaries. Then, using a series of simple tests based on the motion parallax constraint and on depth/motion comparison, we identify a number of edges that can not correspond to motion boundaries. Normal flow values then on both sides of these edges are used to estimate the rotation, in order to subsequently proceed with the main procedure of depth variation minimization. It should be emphasized that the utilization of stereo disparities and the motion parallax constraint will only provide some of depth/motion boundaries. The full map of the motion boundaries arises from minimizing depth smoothness inside the image patches in a computation that performs region growing, and merging-splitting of the patches.

4 Minimizing Depth Variation vs. Epipolar Minimization

The proposed approach delivers the camera motion while at the same time coming up with depth and motion discontinuities. For the sake of the tractability of the exposition we assume in this section that the scene is static and concentrate on the analysis of the novel approach to 3D motion estimation. Many previous approaches have used the assumption of locally smooth (constant, linear, or smoothly varying) scene depth. Without the detection of depth discontinuities, however, an assumption is made which is not valid everywhere in the image. Explicit consideration of the depth discontinuities brings about one fundamental difference. If (and only if) we are able to detect the depth boundaries between surface patches, we no longer need to make smoothness *assumptions*; we are merely utilizing a property of the world. In classical approaches the process of optic flow estimation, which involves smoothing, is separated from the process of 3D motion estimation and structure computation. After optic flow has been fitted to the image data, that is the normal flow, the information about the goodness of the fit is discarded and not considered in the later processes. By combining the processes of smoothing, 3D motion and structure estimation, we utilize this information in all the processes. The estimation of structure and motion thus becomes a geometrical and statistical problem. The challenge lies in understanding how the statistics of the input relates to the geometry of the image information and how to combine the constraints in an efficient way in the development of algorithms.

Consider a small patch P that contains a set of measurements $\dot{\mathbf{r}}_i$ in directions $\mathbf{n}_i$ (normal flows). Given candidate motion parameters $\hat{\mathbf{t}}$ and $\hat{\boldsymbol{\omega}}$ we can estimate, using (2), depth up to an overall scale ambiguity. One possible measure of depth variation is the variance of the depth values, or rather the sum of squared differences of the depth values from a mean $1/\bar{Z}$, i.e.,

$$S_0(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, P) = \sum_i \left(\frac{\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}_i}{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i} - \frac{1}{\bar{Z}} \right)^2$$

For the purpose of relating S_0 to the epipolar constraint, a similar way of writing measure S_0 without explicitly computing depth is:

$$S_0(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, P) = \sum_i W_i (\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}_i - \frac{1}{\hat{Z}} \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i) \qquad (10)$$

where $1/\hat{Z}$ the depth estimate minimizing the measure, not necessarily the mean $1/\bar{Z}$, and $W_i =$

$1/(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2$.

Equation (10) elegantly expresses depth variation inside the patch P while at the same time it has an intuitive geometric interpretation. It is the sum of the squared differences of the normal flow measurements and the corresponding projections of the best flow obtained from the motion parameters, normalized by the estimated translation $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ in order to treat different patches equally. Assuming that the depth is constant in the small patch (fronto-parallel plane), the best inverse depth $1/\hat{Z}$ minimizing S is

$$\frac{1}{\hat{Z}} = \frac{\sum_i W_i(\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)}{\sum_i W_i(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2} \tag{11}$$

If more precision is required, one can model the scene patch by a general plane and use a linear approximation $1/\hat{Z} = \mathbf{z} \cdot \mathbf{r}$ (note that the third component of $\mathbf{r}$ is a constant f, so $\mathbf{z} \cdot \mathbf{r}$ is a general linear function in the image coordinates). Then we have

$$\frac{\partial S_0}{\partial \mathbf{z}} = \sum_i W_i\,(\mathbf{z} \cdot \mathbf{r}_i)(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2 \mathbf{r}_i - \sum_i W_i\,(\dot{\mathbf{r}}_i \cdot \mathbf{n} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}_i)(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)\,\mathbf{r}_i = 0 \tag{12}$$

a set of three linear equations in the three elements of $\mathbf{z}$.

Substituting (11) (or the solution of (12)) into (10), we obtain $S(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, P)$, a function of $\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}$ whose minimum provides the desired 3D motion.

The rest of this section shows that our depth smoothness measure S for a patch P can be decomposed into two components, one component that measures the deviation of the patch from a smooth scene patch and a second component which constitutes a multiple of the epipolar constraint!

Consider the function S_0 in a small image region P. The vectors $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})$ are polynomial functions of image position $\mathbf{r}$ and can usually be approximated by constants within the region. We use a local coordinate system where $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ is parallel to $[1,0,0]^{\mathrm{T}}$. Without loss of generality we can write (in that coordinate system)

$$\begin{aligned} \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) &= [1,0,0]^{\mathrm{T}} \\ \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) &= [u_{\mathrm{rx}}, u_{\mathrm{ry}}, 0]^{\mathrm{T}} \\ \mathbf{n}_i &= [\cos\psi_i, \sin\psi_i, 0]^{\mathrm{T}} \\ u_{\mathrm{n}_i} &= \dot{\mathbf{r}}_i \cdot \mathbf{n}_i \end{aligned} \tag{13}$$

Let us consider the problem of fitting the best constant optical flow $(u_{\mathrm{x}}, u_{\mathrm{y}})$ to the measurements in P, using weights W_i, i.e., minimizing

$$\sum_i W_i \left(u_{\mathrm{n}_i} - (u_{\mathrm{x}}, u_{\mathrm{y}}) \cdot \mathbf{n}_i\right)^2 = \sum_i W_i \left(u_{\mathrm{n}_i} - u_{\mathrm{x}} \cos\psi_i - u_{\mathrm{y}} \sin\psi_i\right)^2 \tag{14}$$

To simplify the notation, we define several sums

$$\begin{array}{rclrcl} S_{\mathrm{cc}} & = & \sum W_i \cos^2\psi_i & S_{\mathrm{uc}} & = & \sum W_i\, u_{\mathrm{n}_i} \cos\psi_i \\ S_{\mathrm{cs}} & = & \sum W_i \cos\psi_i \sin\psi_i & S_{\mathrm{us}} & = & \sum W_i\, u_{\mathrm{n}_i} \sin\psi_i \\ S_{\mathrm{ss}} & = & \sum W_i \sin^2\psi_i & S_{\mathrm{uu}} & = & \sum W_i\, u_{\mathrm{n}_i}^2 \end{array} \tag{15}$$

The vector $(u_{\mathrm{x}}, u_{\mathrm{y}})$ minimizing (14) is obtained by differentiating (14) and solving the following linear system

$$\begin{pmatrix} S_{\mathrm{cc}} & S_{\mathrm{cs}} \\ S_{\mathrm{cs}} & S_{\mathrm{ss}} \end{pmatrix} \begin{pmatrix} u_{\mathrm{x}} \\ u_{\mathrm{y}} \end{pmatrix} = \begin{pmatrix} S_{\mathrm{uc}} \\ S_{\mathrm{us}} \end{pmatrix}$$

We obtain

$$\begin{pmatrix} u_{\mathrm{x}} \\ u_{\mathrm{y}} \end{pmatrix} = \frac{1}{S_{\mathrm{cc}}S_{\mathrm{ss}} - S_{\mathrm{cs}}^2} \begin{pmatrix} S_{\mathrm{ss}} & -S_{\mathrm{cs}} \\ -S_{\mathrm{cs}} & S_{\mathrm{cc}} \end{pmatrix} \begin{pmatrix} S_{\mathrm{uc}} \\ S_{\mathrm{us}} \end{pmatrix} \tag{16}$$

and the minimum error is

$$E_{\mathrm{F}} = S_{\mathrm{uu}} - \frac{1}{S_{\mathrm{cc}}S_{\mathrm{ss}} - S_{\mathrm{cs}}^2}\left(S_{\mathrm{us}}^2 S_{\mathrm{cc}} + S_{\mathrm{uc}}^2 S_{\mathrm{ss}} - 2S_{\mathrm{uc}}S_{\mathrm{us}}S_{\mathrm{cs}}\right) \tag{17}$$

Using the notation (13) we have

$$S_0 = \sum_i W_i \left(u_{\mathrm{n}_i} - u_{\mathrm{rx}} \cos\psi_i - u_{\mathrm{ry}} \sin\psi_i - \frac{1}{\hat{Z}} \cos\psi_i\right)^2 \tag{18}$$

It can be verified that u_{rx} only shifts the best $1/\hat{Z}$, but it does not influence the final measure. Thus we can set u_{rx} to zero without loss of generality and expand S_0 to

$$S_0 = S_{\mathrm{uu}} + u_{\mathrm{ry}}^2 S_{\mathrm{ss}} + \left(\frac{1}{\hat{Z}}\right)^2 S_{\mathrm{cc}} - 2u_{\mathrm{ry}}S_{\mathrm{us}} - 2\frac{1}{\hat{Z}}S_{\mathrm{uc}} + 2\frac{1}{\hat{Z}}u_{\mathrm{ry}}S_{\mathrm{cs}}$$

Minimization of S_0 yields

$$\frac{1}{\hat{Z}} = \frac{S_{\mathrm{uc}} - u_{\mathrm{ry}}S_{\mathrm{cs}}}{S_{\mathrm{cc}}} \tag{19}$$

Let us denote

$$u_{\mathrm{ry}} = u_{\mathrm{y}} + \delta u_{\mathrm{ry}} \tag{20}$$

Measure S is obtained by substituting (19) into S_0. Using (17) it can be written as

$$S = \frac{S_{\mathrm{cc}} S_{\mathrm{ss}} - S_{\mathrm{cs}}^2}{S_{\mathrm{cc}}} \, (\delta u_{\mathrm{ry}})^2 + E_{\mathrm{F}} \tag{21}$$

Considering the epipolar constraint in patch P (equation (6)) and supposing that the flow vector $\dot{\mathbf{r}}$ has been obtained by minimization of (14), we can write it as (u_x, u_y). Substituting into (6), we obtain

$$\frac{((\hat{Z} \times \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})) \cdot (\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})))^2}{\|\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})\|^2} = (u_{\mathrm{y}} - u_{\mathrm{ry}})^2 = (\delta u_{\mathrm{ry}})^2$$

that is, $(\delta u_{\mathrm{ry}})^2$ represents the distance of the estimated flow from the epipolar line, i.e., the quantity optimized in epipolar minimization. Thus (21) shows the geometric meaning of our proposed measure S. The first component in (21) is related to the epipolar constraint and it depends on the 3D motion estimate, as well as the gradient distribution in the patch. The second component in (21), E_{F}, represents how well the scene is approximated by a plane and it is independent of the 3D motion estimate. In classic approaches, after optic flow is computed, the term E_{F} is not considered any further and the estimation of 3D motion parameters is based only on the distance from the epipolar line. Here we keep this term to utilize it for segmentation. In addition, in comparison to other approaches, in our measure the distance to the epipolar line appears mingled up with a function describing the gradient distribution. Thus by studying the statistics of this function insight can be gained which might be exploited in algorithms for 3D motion estimation and segmentation. However, in the current paper we have not delved into such analysis yet; we only qualitatively describe this function and mention that it is closely related to the bias one would obtain when estimating flow on the basis of image derivatives [29].

To summarize, we have obtained for our depth variation measure S inside a patch that $S = C\,(\delta u_{\mathrm{ry}})^2 + E_F$, where C is a factor that depends on the positions and directions of the normal flow measurements in the region, and δu_{ry} is deviation from the epipolar constraint. C can be written as

$$\frac{(\sum_i \cos^2\psi_i)(\sum_i \sin^2\psi_i) - (\sum_i \cos\psi_i \sin\psi_i)^2}{(\sum_i \cos^2\psi_i)}$$

Applying the Cauchy-Schwartz inequality to the numerator we see that $C \geq 0$ and can become zero only if $\sin\psi_1 : \sin\psi_2 : \cdots : \sin\psi_n = \cos\psi_1 : \cos\psi_2 : \cdots : \cos\psi_n$, which happens when all normal flow directions inside the patch are parallel. In fact, C measures the variation of normal flow directions inside the region. The larger the variation of normal flow directions, the larger C is, thus giving more weight to regions where more information is available. At an image patch corresponding to a smooth scene patch, $E_F = 0$, making minimization of S equivalent to epipolar minimization; but for an unsmooth patch where optic flow is not as reliable, $E_F \neq 0$ and this statistical information is utilized to find depth boundaries.

5 The Algorithm

We consider a stereo system in motion. To express the stereo configuration we employ the differential model of rigid motion (Figure 5). Aligning the fixation plane with the XZ plane, the displacement of the left camera with respect to the right camera is described by a translation with velocity $(U_S, 0, W_S)$ and a rotation with velocity $(0, \beta_S, 0)$. Thus, the horizontal disparity d amounts to

$$d = \frac{d_{\text{tr}}}{Z} + d_{\text{rot}} = W_S \frac{x - x_0}{Z} + \beta_S(\frac{x^2}{f} + f)$$

where d_{tr} and d_{rot} are the translational and rotational components of the horizontal disparity and $x_0 = \frac{fU_S}{W_S}$ is the x-coordinate of the epipole of the stereo system (the y-coordinate is zero).

[Figure 5 about here.]

Whenever we need to compare information derived from motion and stereo and disambiguation is necessary, we use the subscripts *motion* and *stereo* for variables corresponding to the respective sources of information.

From the left and right camera images acquired at time t we compute the field of horizontal disparities and from the images acquired by one of the cameras at times t and $t + \delta t$ we compute the normal flow field. As a preprocessing step, depth discontinuities are detected by using the disparity field in conjunction with image intensity and flow field information. The algorithm then proceeds as a search in the space of translational directions:

$\{$ `For each candidate FOE in the search area, do:` $\{$

1. `Check the flow field in the neighborhood of the depth discontinuities to find`

those discontinuities which clearly cannot be only due to depth changes in the background. This is accomplished in two ways:

- by means of the motion parallax constraint
- by comparison of the difference in the flow values in the neighborhood of the discontinuity with the difference in the disparity values.

2. Estimate the rotation from normal flow vectors in the neighborhood of the depth discontinuities which are in accordance with the tested FOE.

3. On the basis of the estimated 3D motion, compute the depth of the scene within image patches which have been found to have similar value in the disparity map. Then, derive the ''smoothness of depth,'' which amounts to the variation of the depth estimates.

4. Extract image patches corresponding to large variations in the depth estimates (i.e., unsmooth areas) (these represent candidate moving objects). Compute an overall measure of smoothness for the remaining areas, which are considered background.

}

The translational direction candidate (and its corresponding rotation computed in step 2) giving rise to the smoothest background, is considered to be the correct 3D motion estimate, and the unsmooth patches are segmented as moving objects. }

The possibility of performing the search in parallel, i.e., computing the "smoothness" of the scene in view for the different 3D motion parameters simultaneously, suggests that the proposed technique could be carried out in real time given sufficient resources. In the next section, the modules of the algorithm are described in detail. In particular, Section 6.1 refers to the preprocessing, amounting to the detection of some depth discontinuities. Sections 6.2 and 6.3 correspond to step 1; Section 6.4 corresponds to step 2; Sections 6.5 and 6.6 correspond to steps 3 and 4 of the algorithm; and Section 6.7 discusses possible further improvements through feedback mechanisms.

Section 7 describes experimental results with synthetic and real sequences and Section 8 summarizes the work.

6 The Modules of the Algorithm

6.1 Detection of the discontinuities

The algorithm starts by detecting discontinuities in the depth map. To avoid the necessity for a dense depth map and to account for the difficulty in estimating accurate disparity measurements, this is achieved by applying an edge detector to the intensity image and then filtering out the edges that correspond to discontinuities in depth.

Edge detectors such as the Canny edge detector [38] used in our implementation give a variety of different types of edges. Some edges correspond to depth discontinuities which are not motion discontinuities (in the remainder of this paper called *depth edges*), some edges correspond to the edges of moving objects (called *motion edges*), and other edges are just *texture edges*: there is no difference between the depth or motion on either side of the edge.

To detect, among all these edges, those corresponding to changes in depth, both stereo and motion information is used:

- The depth is always different on the left and right sides of a depth edge and often different at a motion edge. We therefore measure the difference in horizontal disparity for all the extracted edges, and if this difference is large at an edge, we know it is either a depth or a motion edge, but certainly not a texture edge.

- Further reduction of the number of candidate depth and motion edges is obtained using flow field information. Again, we scan along the left and right sides of each edge and estimate the optical flow for relatively small patches on either side of the edge. (When estimating the optical flow from normal flow we make sure that we do not cross other edges, so that we are guaranteed to stay within smooth patches in which the estimation of optical flow doesn't pose problems.) If the flow estimates in the left and right patches are close in value we disregard the edge, as it probably corresponds to a texture edge or a depth discontinuity, but with a relatively small depth variation, which also should be discarded as such edges tend to make the following motion estimation modules less robust.

The last two steps leave us with a number of edges that can be either depth or motion edges. As a first step in the motion estimation cycle, we eliminate from the edges found those which are clearly motion edges, as is explained in the next two sections: *Motion parallax* and *Depth/motion comparison*.

6.2 Motion parallax

The motion parallax constraint, as is well known [39, 40], provides information about the direction of translation. For general motion sequences with no knowledge available about the scene in view, it is very difficult to utilize this constraint, but if we know the locations of some depth discontinuities it becomes a valuable tool.

Consider a patch with a depth discontinuity that belongs to a static part of the image. For two nearby points $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$, one on each side of the depth discontinuity, consider the optical flow values $\overrightarrow{u_1}$ and $\overrightarrow{u_2}$. As the rotational components of the flow are functions only of the image coordinates, not of the depth of the scene, their values are nearly equal. Thus, subtracting the flow vectors leaves us with a (vector) equation which defines the direction of translation:

$$\overrightarrow{u_1} - \overrightarrow{u_2} = W_{\text{motion}}(\vec{p} - p_{0,\text{motion}}) \left(\frac{1}{Z_1} - \frac{1}{Z_2} \right) \tag{22}$$

From this it can be seen that the focus of expansion lies on a line with orientation $\overrightarrow{u_1} - \overrightarrow{u_2}$ and passing through $\overrightarrow{p_1} \approx \overrightarrow{p_2}$, $\overrightarrow{p} = \overrightarrow{p}_1 + k(\overrightarrow{u}_1 - \overrightarrow{u}_2)$.

As we can also compute from the depth map the ordinal relationship of the depth values at the two sides of the discontinuity, we can further restrict the focus of expansion to a half-line.

When applying the motion parallax constraint to a motion edge, the difference between the computed flow vectors originates from the relative motion between the camera and the moving object, and in general is different from the direction of the camera's translation. This allows us to distinguish between different types of edges.

In the present algorithm we use the motion parallax constraint to verify whether a particular choice of FOE is compatible with the depth discontinuities. The main loop of the algorithm proposes a number of FOEs and the motion parallax method is used to distinguish between motion edges and depth edges: For each of the edges derived earlier, we choose pairs of patches, with one patch to the right and one to left of the edge for which we derive the difference of flow values. To correct for inconsistencies in the flow estimates we allow the estimated direction of translational flow to deviate by a predefined angle from the tested direction specified by the FOE. If all the patches of an edge satisfy the motion parallax constraint, then we consider the corresponding edge to be a depth edge for the tested FOE.

6.3 Depth/Motion comparison

In addition to the previously discussed method, we perform some simple comparisons between motion and stereo measurements to detect those motion edges for which there is a large difference between the 3D motions of the background and the moving object.

Comparing the difference of the optical flow vectors within a patch containing a depth discontinuity (22) with the difference in horizontal disparity

$$d_1 - d_2 = W_{\text{stereo}}(x - x_{0,\text{stereo}}) \left(\frac{1}{Z_1} - \frac{1}{Z_2} \right)$$

we can derive the ratio

$$\frac{\|\overrightarrow{u_1} - \overrightarrow{u_2}\|}{(d_1 - d_2)\,\|p - p_{0,\text{motion}}\|} = \frac{W_{\text{motion}}}{W_{\text{stereo}}\left((x - x_{0,\text{stereo}})\right)}$$

Since for most stereo systems the FOE will be far outside the image boundaries, and thus the distance from image points to the FOE does not vary much within the image, the above ratio for patches with depth discontinuities is nearly constant.

We thus compute for all the detected depth/motion edges the ratio of $\|\overrightarrow{u_1} - \overrightarrow{u_2}\|$ and $(d_1 - d_2)\,\|p - p_{0,\text{motion}}\|$ for patches to the left and right of the edges. Assuming that the majority of these edges correspond to the background, we cluster the computed values and extract the largest cluster [41]. All other edges are disregarded, as they will most likely correspond to motion discontinuities or to patches with erroneous flow or disparity values.

6.4 Rotational estimate

The preliminary filtering of most of the motion edges allows the estimation of rotation. The computations are based on the observation that vectors which are orthogonal to the translational component of the flow are due to rotation only, as explained in Section 3. In particular, the flow component u_n in direction $\vec{\mathbf{n}} = (y_0 - y, x - x_0)$, where (x_0, y_0) is a candidate FOE, provides a function linear in the rotational components. Thus, we locate the normal flow vectors in the direction defined in this way in the image patches close to the extracted depth edges and estimate the three rotational parameters using a least square procedure with outlier rejection. The residual of the fit provides some information as to whether the tested FOE describes the motion field of the background. If it is below a threshold we continue with the next processing step; otherwise, we disregard the FOE candidate.

6.5 Smoothness

Sections 2 and 3 described in detail the geometric intuition behind the minimizing of depth variability for performing 3D motion estimation and segmentation.

On the basis of the tested FOE candidate the smoothness of depth is evaluated within image patches. Assuming that the image patches correspond to smooth scene patches, the correct 3D motion is guaranteed to give rise to the smoothest depth values in the static parts of the scene. Patches with moving objects are generally estimated as unsmooth. Thus, we extract the unsmooth patches, which we consider as corresponding to moving objects, and evaluate the smoothness of the background. The FOE candidate which gives the smoothest value is considered to be the correct 3D motion estimate and moving objects are segmented using a region growing technique on the basis of information about edges, depth and smoothness (Section 6.6).

Certain issues mentioned above need further explanation. As the labeling of the scene, that is, the decision as to which patches are labeled background and which are labeled moving objects, cannot be made without additional knowledge, we assume the background to cover at least half of the image. Only if smoothness above a threshold within at least 50% of the image is found, is the FOE considered to be a candidate for the camera's 3D motion.

To evaluate the smoothness within an image patch, we compute the depth estimates for all image points with normal flow vectors using (7). The standard deviation of the estimates $\hat{Z}$ divided by $\|\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})\|$ (for normalization purposes) is taken as the smoothness measure. As overall smoothness measure we consider the sum of standard deviations within single image patches.

The smoothness measure depends on the actual and estimated motion parameters, as well as the distribution of flow vectors and the scene depth within the image patches.

Equation (21) showed that if the patch P is the image of a smooth scene patch, then since $E_{\rm F}$ would be zero, minimizing S is equivalent to the epipolar minimization. We provide here an additional, more intuitive, simple explanation. Let us approximate the vectors $\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})$, $\mathbf{u}_{\rm tr}(\mathbf{t})$ and $Z\mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega})$ within the patch by constants and assume the normal flow measurements to be error-less. If the image patch corresponds to a smooth scene patch with small variance in the depth values, or $1/Z$ nearly constant, the depth estimates $\hat{Z}$ (from (8)) are also nearly constant if the vectors $\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})$ and $\frac{\mathbf{u}_{\rm tr}(\mathbf{t})}{Z} + \mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega})$ are parallel. Thus the depth variance for the correct 3D motion is among the smallest of all 3D motion estimates.

If the image patch contains a depth discontinuity, the distribution of the normal flow vectors becomes influential. In such cases, incorrect 3D motion estimates may yield the smallest depth variance. For example, consider the two scene patches at the two sides of the discontinuity having depth values Z_1 and Z_2 and let the gradient direction in one scene patch be $\mathbf{n}_1$ and that in the other patch be $\mathbf{n}_2$. Then 3D motion estimates (with $\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega})$) which yield smaller differences in the distortion factors D_1 and D_2 (evaluated at Z_1 and Z_2) than the ones for the correct estimate give rise to a smoother image patch. The smaller the range of gradient directions is, the larger their influence on the 3D motion estimates will be. If the range of gradient directions is large, the correct 3D motion can be expected to yield the smallest depth variance, as can be seen from (8).

Thus, although individual patches with depth discontinuities may not yield the correct 3D motion, in our approach we have available some depth segmentation, and thus a partition into image patches corresponding to smooth scene patches can be guaranteed at many locations. Therefore, the sum of the standard deviations of depth estimates of all patches is a good measure for evaluating 3D motion. However, the main advantage of the introduced constraint lies in the fact that, while 3D motion is evaluated, depth information is also computed which facilitates the segmentation into coherent scene patches. Consider an image patch with a depth discontinuity. The flow vectors in this patch are separated into two regions with two different depth values. Thus, in step 3 of the algorithm, when the smoothness of depth within the image patch is computed, a segmentation algorithm is applied to detect patches with depth boundaries and split them into regions of similar depth values. Segmentation on the basis of depth information is easier than on the basis of raw 2D image measurements, and it further improves the smoothness measure for the correct background motion.

6.6 Region growing

To segment the moving objects, region growing is applied. Image patches in the neighborhood of motion edges which have been found with the motion parallax constraint or the depth/motion comparison and patches which are unsmooth are grown between the detected edges. The region growing is based on a combination of depth, flow, and smoothness estimates; if any of these measurements can be derived, it should vary smoothly within the boundaries of the moving object.

6.7 Feedback

The close coupling of 3D motion estimation and segmentation generally requires iterative computations which are hypothesized to be implemented in nature as feedback mechanisms [42]. That is, it is assumed that biological systems first, in a number of processing steps, perform some preliminary independent motion detection and 3D motion estimation using local and global flow field representations. The derived estimates are then fed back to earlier computational units in order to accurately localize moving objects from local flow field information. In addition, the results of the segmentation can be used to improve the motion estimation. Regarding our algorithm, such feedback mechanisms can be easily realized using the different computational steps described above. In a first run the 3D motion may not be computed accurately and thus the background may not be computed as very smooth. In such cases we can use the obtained segmentation and recompute the rotational parameters from all the background patches found and subsequently the smoothness estimates for a small number of translational estimates.

7 Results

We have tested the proposed algorithm both on synthetic and real sequences. The results for the synthetic sequence are presented in Section 7.1; the results for the real sequence are detailed in Section 7.2.

7.1 Synthetic sequence

A synthetic stereo motion sequence was created using *Open Inventor*, a 3D programming language mainly used on SGI systems. The left image of the first frame is shown in Figure 6. The camera was both translating (with an FOE location as shown in Figure 13) and rotating, and independent motion was due to the box in the lower left corner of the image, moving to the left and rotating.

Figure 7 shows the result of the preprocessing step, the extraction of plausible depth and motion edges. The next four figures display the intermediate results of the four processing steps for the correct FOE. In particular, Figure 8 illustrates the removal of the motion edges by means of the motion parallax constraint and the depth-motion comparison technique (step 1 of the algorithm). Figure 9 shows the fitting of the rotational parameters using least squares with outlier removal (step 2 of the algorithm). For the proposed FOEs the residual of the estimate is displayed as a gray value, with black corresponding to the smallest value; that is, the intensity at a patch denotes the error of fitting rotation for a candidate FOE at the middle of the patch. Locations for which either the residual or the number of outliers was above a threshold and which were not examined further are marked as white. Figure 10 shows the smoothness computed within patches for the correct FOE (step 3 of the algorithm). The patches are adapted in size and shape to the depth boundaries previously found. Figure 11 shows the result of the segmentation using region growing of the unsmooth areas (step 4 of the algorithm). To provide a comparison, in Figure 12 the smoothness of image patches is also shown for a wrong FOE candidate (as marked in Figure 13).

The integrated smoothness (as computed in step 4 of the algorithm) for all the FOEs is displayed in Figure 13. The correct FOE has been found as the best motion candidate (the patch with the darkest gray value). In the implementation the smoothness is actually computed only for FOEs with good rotational fit and unsmooth areas in more than half of the image plane as shown in

Figure 14.

[Figure 6 about here.]

[Figure 7 about here.]

[Figure 8 about here.]

[Figure 9 about here.]

[Figure 10 about here.]

[Figure 11 about here.]

[Figure 12 about here.]

[Figure 13 about here.]

[Figure 14 about here.]

7.2 Real sequence

In this section we show the results of our algorithm on a real sequence taken with a BiSight stereo head. One frame of the sequence is shown in Figure 15. The camera was moving with translation (as shown in Figure 20) and rotation, and independent motion was due to a person walking to the left in the image while at the same time rotating and lifting his arms. The result of the extraction of depth and motion edges is displayed in Figure 16. Next, the different processing steps for the correct FOE candidate are shown. Figure 17 displays depth edges found. Figure 18 shows the smoothness computed within image patches and Figure 19 shows the result of the segmentation. The integrated smoothness for all the proposed FOEs is shown in Figure 20. The correct FOE has been found as the best motion candidate. Here, for illustration purposes, the smoothness is displayed also for FOE candidates which were discarded at earlier steps of the algorithm.

[Figure 15 about here.]

[Figure 16 about here.]

[Figure 17 about here.]

[Figure 18 about here.]

[Figure 19 about here.]

[Figure 20 about here.]

8 Conclusions

The challenge posed by the problem of motion segmentation from visual cues lies in the understanding of the processes guiding the simultaneous estimation of egomotion and moving object detection. In this paper we provided a technique which considers these processes for a stereo system in motion. The basis of the approach lies in the understanding of the interaction between 3D shape and motion estimation, which led to the formulation of the "smoothness constraint." Wrong 3D motion estimates give rise to depth values that are locally unsmooth, i.e., they vary more than the correct ones. This was exploited in order to obtain the 3D motion that locally provides the smoothest depth while recovering moving objects. The technique can be extended to monocular video sequences in a variety of ways. The main problem is to identify some parts of the background, in order to estimate rotation. Possible approaches include rigidity tests—for example, by estimating optic acceleration, spatial color/texture coherence or temporal integration. Alternatively, tests could be performed to rule out image locations as candidate background regions. These include a series of tests based on the depth positivity constraint such as the ones in [43].

References and Notes

[1] Special thanks to Sara Larson for help in preparing this paper, and to Filip Defoort for performing experiments.

[2] A. Bandopadhay and Y. Aloimonos. Image motion estimation by clustering. *International Journal of Imaging Science and Technology*, 2:345–355, 1990.

[3] M. Bober and J. Kittler. Robust motion analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 947–952, 1994.

[4] P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on statistical regularization method in space and time. In *Proc. First European Conference on Computer Vision*, pages 307–311. Springer-Verlag, 1990.

[5] P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera. In *Proc. IEEE Workshop on Visual Motion*, pages 2–12, 1989.

[6] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. Second European Conference on Computer Vision*, pages 282–287. Springer-Verlag, 1992.

[7] J.-M. Odobez and P. Bouthemy. MRF-based motion segmentation exploiting a 2D motion model and robust estimation. In *Proc. International Conference on Image Processing*, volume III, pages 628–631, 1995.

[8] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 520–526, 1997.

[9] J. J. Koenderink and A. J. van Doorn. Facts on optic flow. *Biological Cybernetics*, 56:247–254, 1987.

[10] M. V. Srinivasan. An image interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, 71:401–415, 1994.

[11] M. Black. Recursive non-linear estimation of discontinuous flow fields. In *Proc. Third European Conference on Computer Vision*, pages 138–145. Springer-Verlag, 1994.

[12] G. Adiv. Determining 3D motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:384–401, 1985.

[13] A. F. Bobick. Using stability of interpretation as verification for low level processing: An example from egomotion and optic flow. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 718–719, 1993.

[14] T. Boult and L. Brown. Factorization-based segmentation of motions. In *Proc. IEEE Workshop on Visual Motion*, pages 179–186, 1991.

[15] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proc. International Conference on Computer Vision*, pages 1071–1076, 1995.

[16] R. C. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7:33–46, 1991.

[17] D. Sinclair. Motion segmentation and local structure. In *Proc. Fourth International Conference on Computer Vision*, pages 366–373, 1993.

[18] W. B. Thompson and T.-C. Pong. Detecting moving objects. *International Journal of Computer Vision*, 4:39–57, 1990.

[19] P. H. S. Torr and D. W. Murray. Stochastic motion clustering. In *Proc. Third European Conference on Computer Vision*, pages 328–337. Springer-Verlag, 1994.

[20] Z. Zhang, O. D. Faugeras, and N. Ayache. Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. In *Proc. Second International Conference on Computer Vision*, pages 177–186, 1988.

[21] J. Weber and J. Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2), 1997.

[22] K. J. Bradshaw, P. F. McLauchlan, I. D. Reid, and D. W. Murray. Saccade and pursuit on an active head/eye platform. *Image and Vision Computing*, 12(3):155–163, 1994.

[23] S. M. Smith and J. M. Brady. ASSET-2: Real time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 1995.

[24] T. Uhlin, P. Nordlund, A. Maki, and J.-O. Eklundh. Towards an active visual observer. In *Proc. Fifth International Conference on Computer Vision*, Cambridge, MA, June 1995.

[25] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Proc. Third European Conference on Computer Vision*, pages 316–327. Springer-Verlag, 1994.

[26] C. S. Wiles and M. Brady. Closing the loop on multiple motions. In *Proc. Fifth International Conference on Computer Vision*, pages 308–313, 1995.

[27] Q. F. Zheng and R. Chellappa. Motion detection in image sequences acquired from a moving platform. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 201–204, 1993.

[28] I. J. Cox. A review of statistical association techniques for motion correspondences. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[29] C. Fermüller, R. Pless, and Y. Aloimonos. The Ouchi illusion as an artifact of biased flow estimation. Technical Report CAR-TR-890, Center for Automation Research, University of Maryland, 1998.

[30] The effect can be attained with small retinal motions or a slight jiggling of the paper and is robust over large changes in the patterns, frequencies and boundary shapes.

[31] B. K. P. Horn and E. J. Weldon, Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[32] Due to the scaling ambiguity, only the direction of translation (Focus of Expansion (FOE) or Focus of Contraction (FOC), depending on whether the observer approaches or moves away from the scene) and the three rotational parameters can be computed.

[33] K. Daniilidis and M. E. Spetsakis. Understanding noise sensitivity in structure from motion. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter 4. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.

[34] It is worth noting that the epipolar constraint is independent of the scene depth. This is one of the reasons for its elegant mathematical properties.

[35] L. Cheong, C. Fermüller, and Y. Aloimonos. Effects of errors in the viewing geometry on shape estimation. *Computer Vision and Image Understanding*, 71:356–372, 1998.

[36] S. M. Zeki. *A Vision of the Brain.* Blackwell Scientific Publications, 1993.

[37] C. Fermüller and Y. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995.

[38] J. Canny. Finding edges and lines in images. Technical Report 720, M.I.T. Artificial Intelligence Laboratory, June 1983.

[39] J. H. Rieger and D. T. Lawton. Processing differential image motion. *Journal of the Optical Society of America A*, 2:354–359, 1985.

[40] H. L. F. von Helmholtz. *Handbuch der Physiologischen Optik.* Leopold Voss, Hamburg und Leipzig, 1896.

[41] To also deal with scenes containing large numbers of moving objects, we compare the classification obtained in this way with the results obtained from the motion parallax constraint, and in cases of large discrepancy we ignore the depth/motion comparison.

[42] C. Fermüller, R. Pless, and Y. Aloimonos. Families of stationary patterns producing illusory movement: Insights into the visual system. *Proc. Royal Society, London B*, 264:795–806, 1997.

[43] C. Fermüller and Y. Aloimonos. On the geometry of visual correspondence. *International Journal of Computer Vision*, 21:223–247, 1997.

List of Figures

Figures

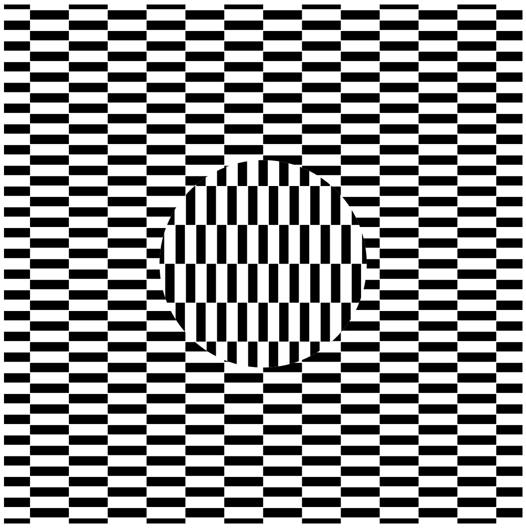


Figure 1: A pattern similar to one by Ouchi.

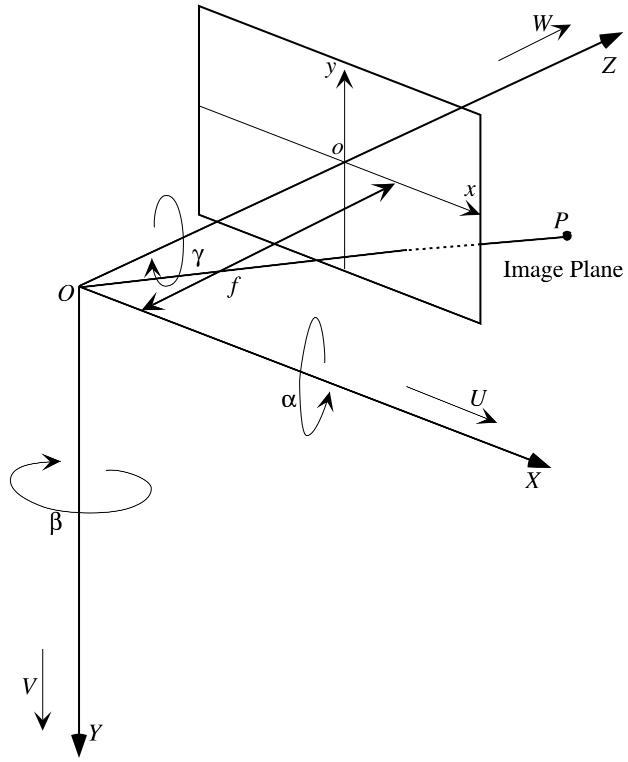
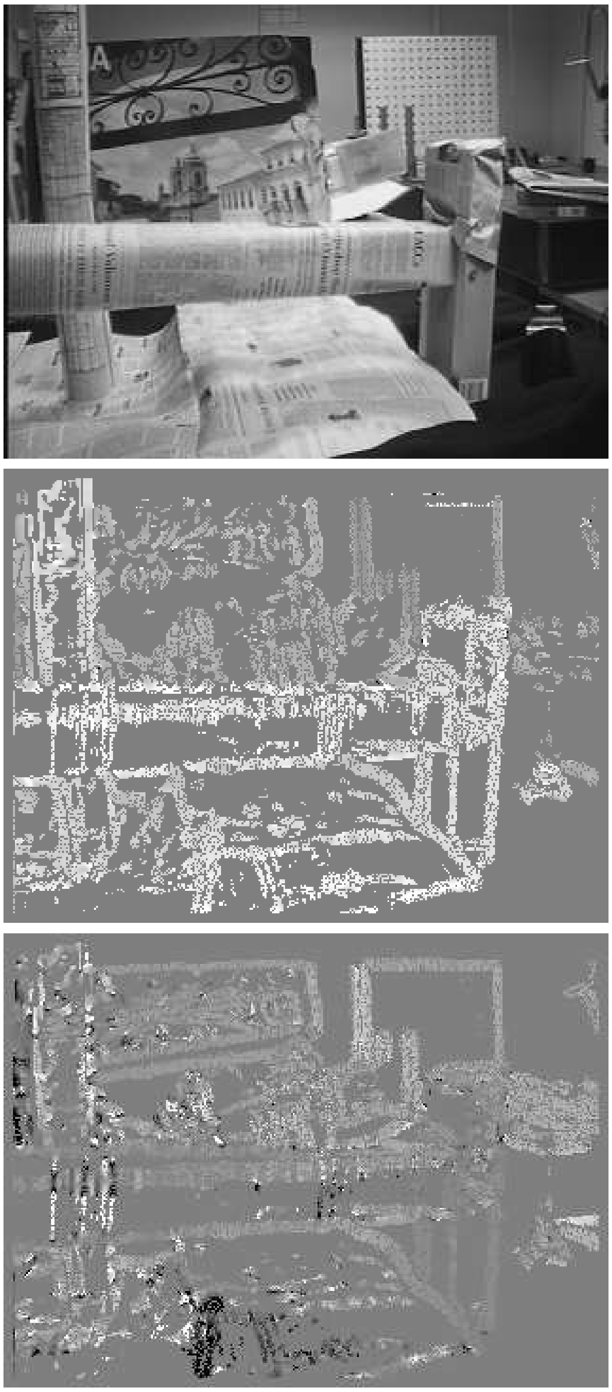


Figure 2: The image formation model for motion.



(a)

(b)

(c)

Figure 3: (a) One input frame of the lab sequence. (b) Inverse depth estimated for the correct FOE ((397, −115) pixels from the image center). (c) Inverse depth for an incorrect FOE ((−80, 115) pixels from image center). The gray-level value represents inverse estimated depth with mid-level gray shown in places where no information was available, white representing positive $1/\hat{Z}$ and black representing negative $1/\hat{Z}$.

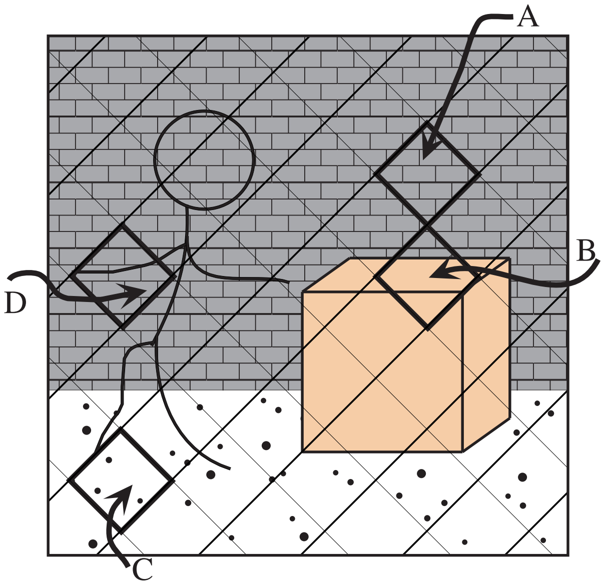


Figure 4: An image showing one frame of a hypothetical sequence acquired by a moving camera. Note the human moving on the path (dotted region) in front of a wall and the stationary cube on the right. The image is divided into small patches (diamonds) where smoothness of depth is evaluated on the basis of candidate 3D motion. Patches A and C corresponding to background will produce small measure of depth variation for the correct motion. Patch B will produce large depth variation as it contains depth discontinuities but after it is split into two patches will produce small variation of depth in each patch. Patch D will give an unsmooth (rugged) depth function. Localization of the moving boundaries is achieved by applying several constraints (see Sections 6.1–6.7).

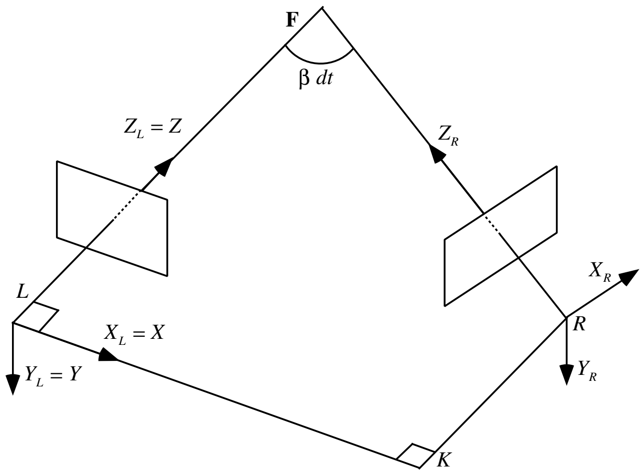


Figure 5: Binocular viewing geometry. $LK = U_S\, dt$ (translation along the X axis), $KR = W_S\, dt$ (translation along the Z axis), $LFR = \beta_S\, dt =$ convergence angle (resulting from rotation around the Y axis). L, K, R, F are in the fixation plane and dt is a hypothetical small time interval during which the motion taking $X_L Y_L Z_L$ to $X_R Y_R Z_R$ takes place.

Figure 6: Left image of first frame for the synthetic scene.

Figure 7: Extracted motion and depth discontinuities.

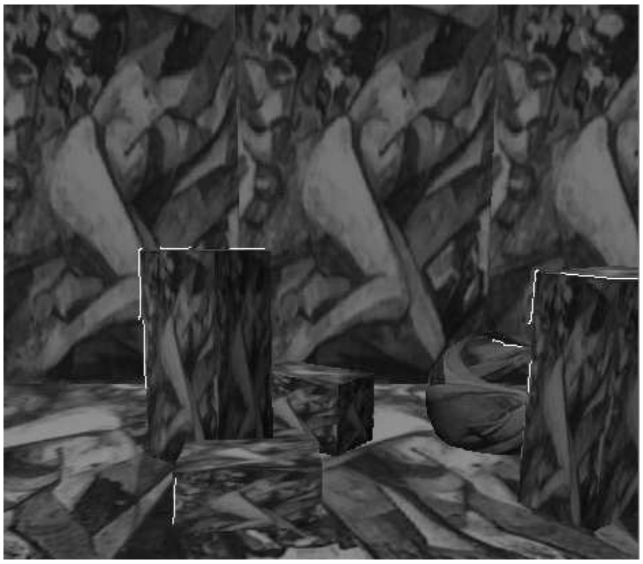


Figure 8: Extracted depth discontinuities for the approximately correct FOE.

Figure 9: Fitting of the rotation for the approximately correct FOE. The gray level denotes the residual with black corresponding to the smallest value. FOEs where the residual and/or number of outliers in the rotation fitting process is too large are marked by white boxes. The rectangle denotes the location of the image.

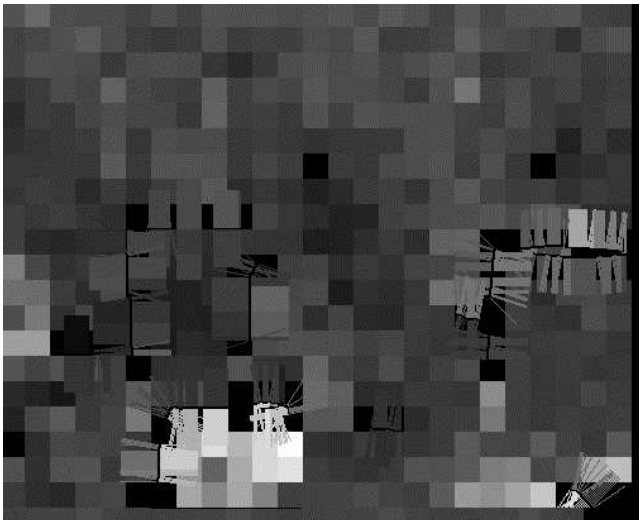


Figure 10: Estimated smoothness for the approximately correct FOE. The gray level decodes the value of the measure of smoothness (i.e., the standard deviation of the depth estimates) within an image patch with black corresponding to the smallest and white to the largest value.

Figure 11: Left image brightened inside the moving region, darkened elsewhere.

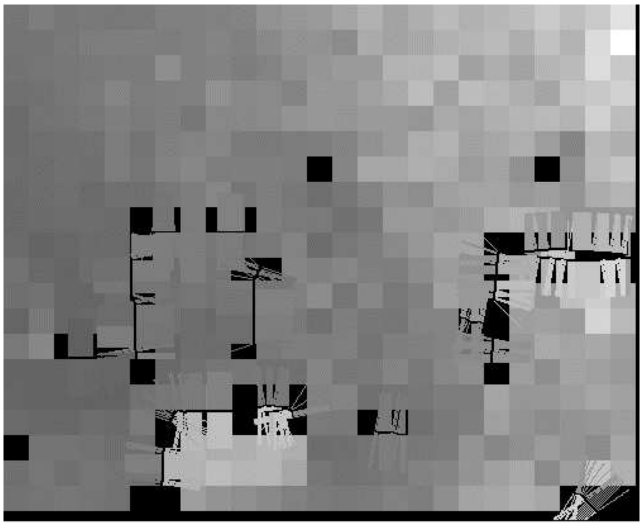


Figure 12: Estimated smoothness for a wrong FOE.

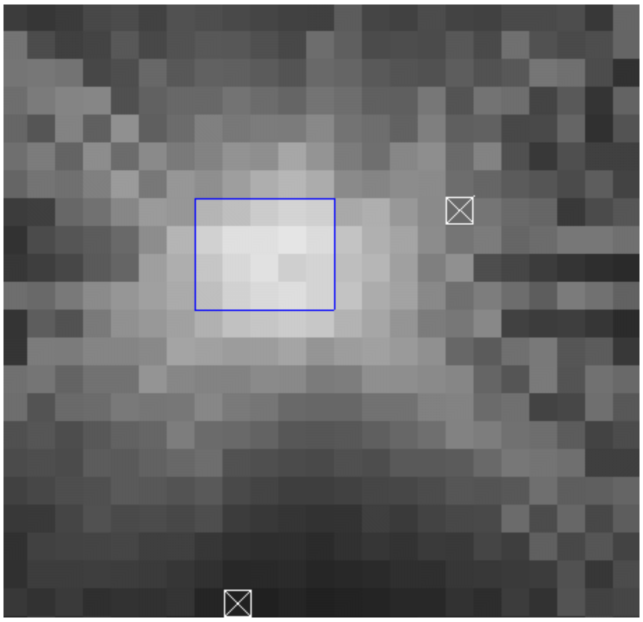


Figure 13: Integrated smoothness value for all proposed FOEs. The image boundaries are indicated by the black rectangle. The correct FOE is indicated by the white box in the lower half of the image; the wrong FOE used for Figure 12 is indicated by the other white box.

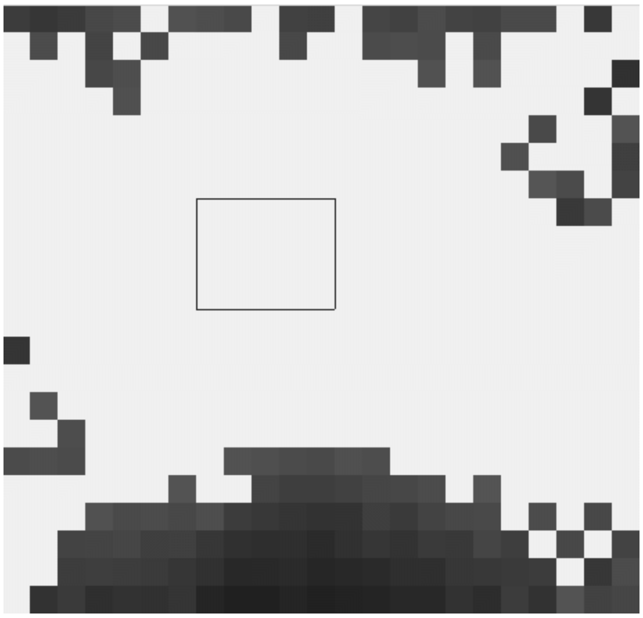


Figure 14: Integrated smoothness value for all proposed FOEs. FOE candidates which were discarded during earlier processing steps are indicated in white.

Figure 15: Left image for the real scene.

Figure 16: Extracted motion and depth discontinuities.

Figure 17: Extracted depth discontinuities for the approximately correct FOE.

Figure 18: Estimated smoothness for the approximately correct FOE with the gray level decoding the value of the measure of smoothness (i.e., the standard deviation of the depth estimates) within an image patch (black=smallest, white=largest value). Note how the independently moving person clearly stands out.

Figure 19: Left image brightened inside the region, which has been detected as independently moving, and darkened elsewhere.

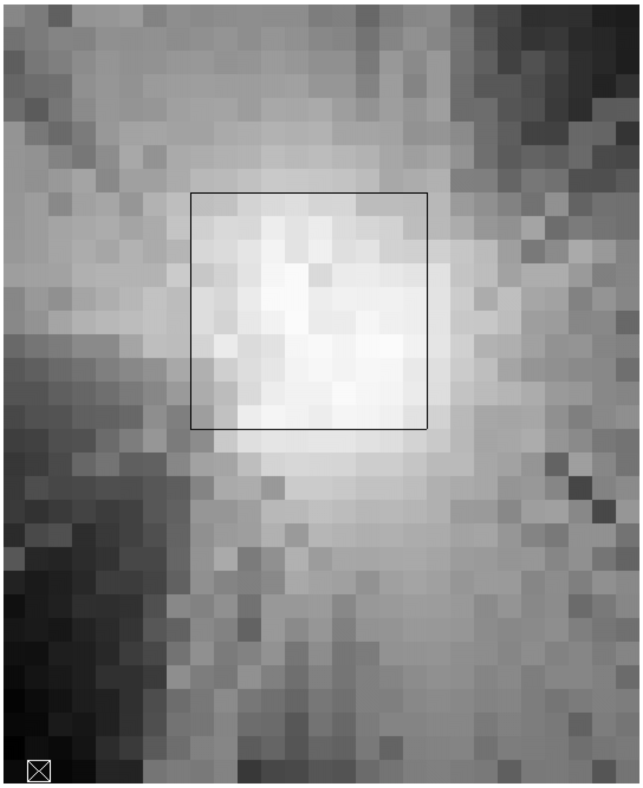


Figure 20: Integrated smoothness value for all proposed FOEs. The image boundaries are indicated by the black rectangle. The correct FOE is marked by the white box.