Final version for the IEEE PAMI Special issue on Digital libraries: Representation and Retrieval

COMPACT REPRESENTATIONS OF VIDEOS THROUGH DOMINANT AND MULTIPLE MOTION ESTIMATION

Harpreet S. Sawhney[1]

Image Information Research Group
David Sarnoff Research Center
CN5300, Princeton, NJ 08543
Email: sawhney@sarnoff.com

Serge Ayer[2]

Signal Processing Laboratory
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
Email: Serge.Ayer@lts.de.epfl.ch

April 11, 1996

Abstract

An explosion of on-line image and video data in digital form is already well underway. With the exponential rise in interactive information exploration and dissemination through the World-Wide Web (WWW), the major inhibitors of rapid access to on-line video data are costs and management of capture and storage, lack of real-time delivery, and non-availability of content-based intelligent search and indexing techniques. The solutions for capture, storage and delivery maybe on the horizon or a little beyond. However, even with rapid delivery, the lack of efficient authoring and querying tools for visual content-based indexing may still inhibit as widespread a use of video information as that of text and traditional tabular data is currently.

In order to be able to non-linearly browse and index into videos through visual content, it is necessary to develop authoring tools that can automatically separate moving objects and significant components of the scene, and represent these in a compact form. Given that video data comes in torrents — almost a megabyte every 30th of a second — it will be highly inefficient to search for objects and scenes in every frame of a video. In this paper, we present techniques to automatically derive compact representations of scenes and objects from the motion information.

Image motion is a significant cue in videos for the separation of scenes into their significant components and for the separation of moving objects. Motion analysis is useful in capturing the visual content of videos for indexing and browsing in two different ways. First, separation of the static scene from moving objects can be accomplished by employing *dominant* 2D/3D motion estimation methods. Alternatively, if the goal is to be able to represent the fixed scene too as a composition of significant structures and objects, then *simultaneous* multiple motion methods might be more appropriate. In either case view-based summarized representations of the scene can be created by video compositing/mosaicing based on the estimated motions. We present robust algorithms for both kinds of representations: (i) dominant motion estimation based techniques which exploit a fairly common occurrence in videos that a mostly fixed background (scene) is imaged with or without independently moving objects, and (ii) simultaneous multiple motion estimation and representation of motion video using *layered* representations. Ample examples of the representations achieved by each method are included in the paper.

[1]This work was performed while this author was a Research Staff Member in the Machine Vision Group at the IBM Almaden Research Center, San Jose, CA.

[2]This work was performed while this author was visiting the IBM Almaden Research Center, San Jose, CA. He was also supported by Thomson-CSF, Rennes, France.

1 Introduction

An explosion of on-line image and video data in digital form is already well underway. With the exponential rise in interactive information exploration and dissemination through the World-Wide Web (WWW), the major inhibitors of rapid access to on-line video data are costs and management of capture and storage, lack of real-time delivery, and non-availability of content-based intelligent search and indexing techniques. The solutions for capture, storage and delivery maybe on the horizon or a little beyond. However, even with rapid delivery, the lack of efficient authoring and querying tools for visual content-based indexing may still inhibit as widespread a use of video information as that of text and traditional tabular data is currently.

There are two important problem domains in video indexing and retrieval where automated video analysis techniques play a role. One domain deals with the creation of compact visual representations of numerous frames in videos through segmentation of scenes, significant structures in scenes and moving objects. By exploiting the redundancy in contiguous frames in videos, and describing the change between frames using a relatively small number of parametric models, visual representations of scenes and objects can be created. The second important domain of problems is in search, recognition and indexing of objects and scenes through queries with visual attributes. The requirement here is to create compact (not necessarily visual) representations in terms of collections of multidimensional features so that objects and scenes can be searched for and recognized efficiently through indexing. The output of visual representations from the first problem domain can help considerably the process of creation of indexable representations by transforming the raw frames of pixels data into a more manageable form. This paper focuses on techniques for compact visual video representations but does not deal with the search and recognition issues directly.

One important step towards automatic annotation and visual content-based index generation for video data is its representation in a compact form so that browsing and search may be made efficient. Videos may be queried and browsed based on their static and dynamic pictorial content. The pictorial content of videos can be succinctly captured by creating static and dynamic representations of the fixed scene (the background), the moving objects and the camera operations/motion. In a future system, these representations should be able to answer queries like "give me shots[3] of a panning camera that capture views of the Golden Gate bridge and the San Francisco skyline, in

[3] A shot is a contiguous set of frames in a video clip depicting a possibly smoothly changing scene captured using a single kind of camera operation or motion.

which a ship was crossing under the bridge" ! Using ideas from static image querying systems like QBIC (Query-By-Image-Content) [29] and Photobook [32], it is conceivable that this query may be transformed into a purely syntactic specification in terms of a painted example of the scene showing the approximate colors [3], sketches [17] and motions [15]. However, in spite of this simplification, matching the "cartoon" version of the query with the actual video data may still be a daunting task if the query needs to be matched to almost every frame. Therefore, representations of appearance, location and motion of significant scene structures and objects need to be created at the time of database creation, and stored in indexable structures so that complex visual queries may be answered. Furthermore, a visually compact and indexed representation of videos will take video browsing beyond the confines of VCR-like linear control functions and frame-at-a-time narrow field of view displays.

The simplest way to represent videos for browsing and querying is through key frames where a key frame is a representative frame in a shot, typically the first, middle, last frame, or a combination of these. In a preliminary demonstration of video querying by visual content in the spirit of the QBIC system, we have used key frames as static image representations of the video on which QBIC-like queries can be performed [11]. However, for video representation, this is just the beginning. The dynamic information contained in videos due to camera and object motion can be exploited to separate the fixed scene from moving objects, and to delineate significant structures in the scene that may be semantically important. In this paper, we present techniques for automatic decomposition of a video sequence into multiple motion models and their layers of supports, which together constitute a compact description of significant scene structures. This includes separation of the dominant background scene from moving objects (the foreground), and representation of the scene and moving objects into multiple layers of motion and spatial support. Furthermore, it is demonstrated how motion-based decomposition of videos can be used to create compact views of numerous frames in a shot by video mosaicing [19, 23, 36, 40, 41, 42]. Once a single image or a few images, that represent all of the significant scene that has been seen in the shot, are authentically created, and the various significant scene components and objects have been separated, these representations can be used for querying based on static image features like color, texture and shape of objects/surfaces and scenes [29, 32], as well as for camera operation based querying. Furthermore, the created representations are an effective tool for video browsing since the visual content of complete shots is compactly represented.

In order to compactly represent the visual content of videos, it is necessary to describe the image motion of all the frames in terms of motions of a relatively small number of patches, each of which can be described by low dimensional parametric motion models. In many real scenarios, it is adequate to describe the static scene, whose image motion is only due to camera motion, using a single global 2D/3D motion model called the *dominant* motion. Dominant 2D models suffice when camera centers do not move appreciably or when the image motion can be well approximated by that of a single plane, and 3D models are necessary when there is significant residual motion parallax beyond a global 2D model. Any departures from the single 2D/3D model, for instance moving objects, are discovered as outliers. We formulate the problem of dominant motion computation as that of model-based robust maximum likelihood estimation (M-estimation) with direct, multi-resolution methods. Two-dimensional (2D) affine and projective parametric models in a robust framework have been used in the past [6, 8, 31] for this problem. We extend this approach along three significant directions. First, our algorithm employs an automatic computation of a *scale parameter* that is crucial in rejecting the non-dominant components as outliers, in contrast with ad-hoc thresholds [20] or predefined schedules for scales [8, 31] used by researchers earlier. Second, in addition to 2D affine and plane projective models for describing image motion using direct methods, we also employ a true 3D model of motion and scene structure imaged with uncalibrated cameras. This model parameterizes the image motion as that due to a planar component and a parallax component [22, 35, 38]. Finally, the effectiveness of image warping over a video sequence using the computed transformation parameters is demonstrated through image registration and mosaicing for both the 2D and 3D models.

Multiple motion models and their support layers can be sequentially fleshed out by applying dominant motion estimation repeatedly. However, the sequential method may not be adequate when no single dominant 2D model is present, for instance in the scene in Fig. 9, and when a 3D model does not afford a compact representation due to explicit representation of depth. Therefore, we present a method for the *simultaneous* estimation of multiple 2D parametric models and their layers of support, called a layered representation[4]. The three major issues in layered motion representation are: (i) how many motion models adequately describe image motion, (ii) what are the motion model parameters, and (iii) what is the spatial support layer for each motion model. We

[4]In [1], a layered representation is used to make the ordinal relations amongst surfaces and objects explicit. Our use of the term implies regions of support and is similar to the one used in [10]. If motion is to be used to derive the ordinal layers, then an algorithm like ours provides a key component from which the ordinal relations can be derived.

formulate the multiple model estimation problem as robust maximum-likelihood estimation (MLE) of *mixture model* parameters and of the layers of support represented as ownership probabilities. The estimation uses a modified Expectation-Maximization (EM) algorithm with the dominant motion estimation as one of its components. The adequate number of models is automatically decided using the minimum description length (MDL) principle that minimizes the encoding length of the model parameters and of the MLE residuals. Furthermore, outliers, that are a problem in a mixture model formulation also, are detected within the EM-MDL framework.

In summary, we comprehensively address the problem of visual scene representations in videos using motion analysis. In situations where global 2D/3D models are adequate in capturing the image motion over time, the dominant motion approach provides an automatic method for computing the model parameters and creating expanded-view visual scene representations. In order to represent the scene as seen in a video as a collection of "objects" with their appearances, it is necessary to derive their layers of support automatically. Our solution combines the advantages of direct motion estimation methods, robust estimation, MDL coding, and mixture of models. We demonstrate that a formulation and an algorithm that integrates all these aspects leads to an automatic layered representation for a variety of videos without any ad-hoc parameters.

2 Background

There are two major approaches to the problem of separating image sequences into multiple significant scene structures and objects based on motion. One set solves the problem by letting multiple models *simultaneously* compete for the description of the individual motion measurements, and in the second set, multiple models are fleshed out *sequentially* by solving for a dominant model at each stage. Model based motion analysis may involve 2D and/or 3D models. Once the image motion models and their layers of support have been computed, video mosaics may be created to compactly represent numerous frames in a shot using one or only few frames. We now review related work for the problems of simultaneous multiple motion and dominant motion estimation, 3D models in motion analysis, and video mosaicing.

Simultaneous Multiple Motion Estimation

Wang and Adelson [43] addressed the problem as the computation of 2D affine motion models and their binary support layers. The ordering in depth is also computed over multiple frames. The essential idea in their work is that of iteratively clustering motion models computed using

pre-computed dense optical flow. The main drawbacks of the above approach are its use of optical flow as an input representation and clustering in the parameter space. In computing optical flow, algorithms generally make smoothness assumptions that can distort the structure of image motion. Second, clustering in the parameter space is generally sensitive to the number of clusters specified. Decisions made for clustering parameter vectors based on distances in the parameter space can lead to clustered parameters that do not describe some valid data well.

Darrell and Pentland too have addressed the problem of multiple motions and support layers estimation within a robust M-estimation and MDL framework [10]. However, their focus has been on correctly estimating the number of models for a sequence rather than on the accuracy and appropriateness of a particular model. Therefore, they have not touched upon the important problems of precision in layer ownerships, accuracy of the motion models, and automatically labeling as outliers those pixels that are not described well by any of the models. They use a truncated quadratic optimization function, that reduces the weight of residuals beyond a threshold to zero. However, the important problem of automatically estimating the threshold (or the scale parameter of the influence function [25]) for the truncated quadratic is not addressed. Furthermore, it is important to detect outliers that are atypical of the complete mixture of models, otherwise the multiple model formulation too may suffer severely from the presence of outliers that may skew the motion and layer estimates.

Hsu et al. [18] in their work on optical flow computation using motion layers have laid out only a qualitative framework. Their algorithms are similar to the ones described above. Our work formally addresses all the issues discussed by them. MacLean et al. [26] addressed the problem of multiple 3D motion segmentation and estimation using the EM algorithm. However, they did not address the problem of automatic determination of the appropriate number of models. Their results were shown using 2D affine flow computed in hand-selected regions. Jepson and Black [21] applied the EM algorithm using the mixture model formulation to compute optical flow without estimating the number of models.

Dominant Motion Estimation

Sequential application of dominant motion estimation methods have been proposed for extracting multiple motions and layers. However, with only few exceptions, almost all the methods have not demonstrated convincingly a complete multiple motion representation of a video using the

dominant motion methods. Of course, as pointed earlier these methods are useful in their own right when the separation of the dominant component itself is useful.

Irani et al. [20] addressed the problem of detecting and tracking multiple moving objects over image sequences through least-squares dominant motion estimation. They do not have any mechanism for automatically estimating the scale parameter of residual errors. Instead, to decide if the point belongs to the model, they apply an absolute threshold to a motion measure defined using normal flow computed at each point.

Ayer et al. [6] too adopted a sequential approach to fleshing out multiple models of motion over an image sequence. In contrast with Irani et al., they applied robust estimators for motion estimation, formulated the problem in terms of time-varying parameters over multiple frames, and combined intensity based segmentation with the motion information. However, they noticed that, even with the use of robust estimators, the sequential dominant motion approach may be confronted with the absence of dominant motion. In this case, no single layer is dominant in its support, in which case the sequential algorithms may need a technique for clustering the sequential support into different support layers. This problem is in itself a difficult task. Another problem is that sequential methods may fail to delineate similar motions into different layers because of the lack of competition amongst the motion models.

Black and Anandan [8], and Odobez and Bouthemy [31] incorporated robust M-estimators in their solution to dominant motion estimation. However, they choose an ad-hoc, pre-defined value of the initial scale and also a schedule for its reduction by a fixed factor through the iterative optimization process. Bober and Kittler [9] combined a Hough technique with robust methods to extract multiple motions through successive separation of dominant motions. Odobez and Bouthemy [30] use dominant motion compensation embedded in a contextual MRF framework to detect multiple motions.

3D Models for Motion Analysis

The use of 3D motion and structure models in model based direct estimation framework was introduced by Hanna [13]. However, this was limited to parameterization of motion and structure for calibrated cameras. In the context of video databases, calibration information may not be available, and may be unnecessary because a full Euclidean reconstruction of motion and structure is not required for video representation. An interesting parameterization for 3D projective structure

and view transformation with respect to a reference view and an arbitrary reference plane (*plane plus parallax*) has recently been introduced in [14, 22, 35, 38]. Shashua and Navab [38], and Hartley [14] use point correspondences to solve for the 3D parameters. Kumar et al. [23] and Sawhney [36] used direct methods to compute the planar transformation and the parallax vectors (see also [39]). We extend the robust M-estimation method to the 3D model [36], thus allowing for separation of outliers, for instance, independently moving objects in an otherwise rigid 3D environment with significant depth variations.

Video and Image Mosaicing

Image mosaicing as a means to obtain a single view representation of a video shot has been proposed by a few researchers [27, 39, 42, 41]. We demonstrate that both the 2D and 3D motion estimates when applied for image warping with temporal filtering can be used to create a mosaiced representation [36]. A similar application with different techniques has recently been presented independently in [19, 23].

3 Plan of the Paper

We first present our work on model-based robust dominant motion estimation using M-estimators. A Gauss-Newton formulation leading to an iterated reweighted least squares algorithm is described in Section 4. Section 5 presents the application of the technique with 2D parametric models to dominant motion separation and video mosaicing. The 2D models do not imply that the scene or motion is 2D, but that the image motion is well approximated by 2D parametric transforms. This approximation turns out to be valid in a large class of real videos and movies where camera pans, tilts, zooms and similar operations are the cause of dominant motions, and when at frame rate the change between consecutive frames is small. Section 6 first presents our formulation and model for 3D parameter estimation, and subsequently shows the results of applying this to 3D scenes and also demonstrates video mosaicing with the automatic computation of the 3D representation.

The remaining sections of the paper are devoted to a description of simultaneous multiple motion and layer estimation. Robust 2D M-estimation is also a key component of our layered representation algorithm. In our formulation, the computed layered representation of motion is the result of optimizing an objective function: the total encoding length of the motion model parameters, of the layers of support, and of the residuals at each pixel resulting from the difference

between the reference intensity map and the intensity map that is warped in accordance with the motion parameters. The optimization is divided into two major steps that are alternated: ML estimation of the motion parameters and layers of support given the number of models, and a greedy incremental strategy for choosing an adequate number of models using the total encoding length given the ML estimates.

The description of the simultaneous multiple motion algorithm starts in Section 8 with a formulation for multiple motion models in terms of mixture models. Given that the image motion can be modeled as caused by a mixture of a *fixed* number of models, the motion model parameters and the ownership probabilities (layers) for each pixel can be computed using an iterative Expectation-Maximization (EM) algorithm; E-step to solve for layers given the motion parameters, the variances, and the model proportions, and the M-step for solving for the latter given the layers. The dominant motion estimation algorithm is used to compute the motion parameters in the EM algorithm. The motion descriptors used are 2D parametric models: translational, affine and projective. Section 9 presents our solution to the problem of determining an adequate number of models using MDL encoding. The subsequent section is devoted to the details of the complete algorithm. Section 11 presents experimental results for the layered motion estimation algorithm.

4 Robust Estimation of a Motion Model using Direct Methods

Our robust formulation combines standard M-estimation [25], with automatic scale computation [34], and direct model-based image motion estimation [7]. Given two images, their motion transformation is modeled as

$$I(\mathbf{p}, t) = I(\mathbf{p} - \mathbf{u}(\mathbf{p}; \boldsymbol{\theta}), t - 1), \tag{1}$$

where $\mathbf{p}$ is the 2D vector of image coordinates, and $\mathbf{u}(\mathbf{p}; \boldsymbol{\theta})$ is the displacement vector at $\mathbf{p}$ described using a parameter vector $\boldsymbol{\theta}$. In the case of 2D global parametric models, a single low-dimensional $\boldsymbol{\theta}$ describes the motion. For the 3D model $\boldsymbol{\theta}$ consists of both a low-dimensional global parametric part, and a projective depth part, one depth parameter per pixel. In order to compute motions of varying magnitudes, the images are represented at multiple scales using Gaussian or Laplacian pyramids. In the following it is assumed that I refers to any of these filtered representations of an original image.

In the M-estimation formulation (that includes sum of squares as a special case), the unknown parameters are estimated by minimizing an objective function of the residual error. In particular,

the following minimization problem is solved:

$$\min_{\boldsymbol{\theta}} \sum_i \rho(r_i;\sigma), \quad r_i = I(\mathbf{p_i},t) - I(\mathbf{p_i} - \mathbf{u}(\mathbf{p_i};\boldsymbol{\theta}), t-1), \tag{2}$$

where $\rho(r;\sigma)$ is the objective function defined over the residuals, r, with a given scale factor, σ, and where i is the index of the ith image pixel. We use two different ρ functions, the sum of squares function and the *Geman-McLure* function:

$$\rho_{SS}(r;\sigma) = \frac{1}{2}\frac{r^2}{\sigma^2}, \qquad \rho_{GM}(r;\sigma) = \frac{\frac{r^2}{\sigma^2}}{1+\frac{r^2}{\sigma^2}}.$$

The ρ_{GM} function gives *non-zero* descending weights that are controlled by the scale parameters. This is more desirable than the behavior of Andrew's sine wave and Tukey's Biweight [12] functions which reduce the weights to zero beyond a threshold.

4.1 Gauss-Newton Formulation for M-estimation

Instead of solving the non-linear system of equations that corresponds to the necessary conditions for a minimum in (2), we apply the Gauss-Newton (GN) method directly to the minimization problem. In the GN method, a descent direction is computed using the gradient and a first order approximation to the Hessian for the given objective function. Writing the $\mathbf{g}$ and $\mathbf{H}$ in terms of ρ and r_i, we get,

$$\mathbf{g}_k = \sum_i \frac{\partial \rho}{\partial r_i}\frac{\partial r_i}{\partial \boldsymbol{\theta}_k} \qquad \mathbf{H}_{kl} = \sum_i \frac{\partial^2 \rho}{\partial r_i^2}\frac{\partial r_i}{\partial \boldsymbol{\theta}_k}\frac{\partial r_i}{\partial \boldsymbol{\theta}_l}, \tag{3}$$

as the kth and the klth elements of $\mathbf{g}$ and $\mathbf{H}$.

With the non-quadratic ρ's, $\frac{\partial^2 \rho}{\partial r_i^2}$ can be negative, therefore one may not get a descent direction. If $\ddot{\rho}(r)$ is approximated by its secant approximation [37, pg. 652], $\frac{\dot{\rho}(r)}{r}$, which is positive everywhere, then the GN equations become,

$$\sum_l \sum_i \frac{\dot{\rho}(r_i)}{r_i}\frac{\partial r_i}{\partial \boldsymbol{\theta}_k}\frac{\partial r_i}{\partial \boldsymbol{\theta}_l}\delta\boldsymbol{\theta}_l = -\frac{\dot{\rho}(r_i)}{r_i} r_i \frac{\partial r_i}{\partial \boldsymbol{\theta}_k}, \tag{4}$$

for $k, l = 1 \ldots K$, and $i = 1 \ldots N$. By comparison with the SS normal equations, it is apparent that the corresponding equations for the robust estimators are simply weighted normal equations with the weight for each measurement i being $\frac{\dot{\rho}(r_i)}{r_i}$.

The plots of the ρ functions for $\sigma = 1.0$, and those of the weights (as in (5)), $\frac{\dot{\rho}(r)}{r}$, are shown in Fig. 1.

$$\frac{\dot{\rho}_{SS}(r)}{r} = \frac{1}{\sigma^2} \qquad \frac{\dot{\rho}_{GM}(r)}{r} = \frac{2\sigma^2}{(\sigma^2 + x^2)^2} \tag{5}$$

It is apparent from the plots that whereas ρ_{SS} weights residuals of all magnitudes uniformly, ρ_{GM}, governed by σ, decreases the influence of large residuals on the solution rapidly.

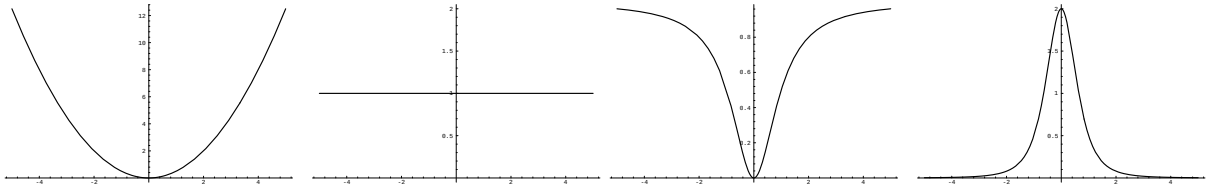

Figure 1: **Left :** Quadratic $\rho(r)$ and the corresponding weight function, $\frac{\dot{\rho}(r)}{r}$. **Right:** Geman-McLure $\rho(r)$, and the corresponding $\frac{\dot{\rho}(r)}{r}$.

4.2 Automatic Scale Estimation and the Hierarchical Algorithm

With the use of M-estimators for dominant motion estimation, a possibility is to use the assumption that the corresponding *pdf*s model the underlying distribution of residuals, and compute the scale, σ, corresponding to the chosen distribution. However, this will lead to complexity in the estimation of σ. Alternatively, it is reasonable to model the residuals using a contaminated Gaussian distribution, where the residuals for the non-dominant components are the contaminants or outliers. Now a robust method for computing the σ of the Gaussian is to be found so that the outliers do not significantly influence the estimate of σ. There are two possibilities for computing σ: (i) as the root mean square sample standard deviation estimate of the weighted residuals, with the weights being $\dot{\rho}(r_i)/r_i$ as in the GN estimation of the parameters, and (ii) as an estimate derived from the median value of the absolute residuals. The latter option is a more robust estimate and is adopted in our experiments. Given contaminated random samples from a zero-mean Gaussian distribution with a given σ, a robust estimate of σ is related to the samples through [34, pg. 202]

$$\sigma = 1.4826\, median_i\, |r_i|.$$

This follows from the fact that the median value of the absolute values of a large enough sample of unit-variance normal distributed one-dimensional values is $0.6745 = 1/1.4826$. The median based estimate has excellent resistance to outliers; it can tolerate almost 50% of them, and can be efficiently computed with a linear time median-finding algorithm.

Given the GN formulation and the step for σ estimation, we embed these in a hierarchical coarse-to-fine direct method [7]. Starting at the coarsest level, given an initial estimate of the parameters $\boldsymbol{\theta}^{(m)}$, typically chosen to represent zero motion at the very start, image $I(\mathbf{p}, t-1)$ is warped so that $I^w(\mathbf{p}, t-1; \boldsymbol{\theta}^{(m)}) = I(\mathbf{p} - \mathbf{u}(\mathbf{p}; \boldsymbol{\theta}^{(m)}), t-1)$. At this step, the residual r at $\mathbf{p}$ is defined as

$$r = I(\mathbf{p} + \delta\mathbf{u}(\mathbf{p}; \boldsymbol{\theta}), t) - I^w(\mathbf{p}, t-1; \boldsymbol{\theta}^{(m)}), \tag{6}$$

where $\delta\mathbf{u}$ is a small unknown increment in $\mathbf{u}$. The robust σ estimate is computed using the residuals r's defined over all $\mathbf{p}$'s. A GN step is then performed with the chosen ρ function to compute a new GN direction $\delta\boldsymbol{\theta}^{(m)}$ using (4) with r as in (6). A line minimization along this direction is performed to get the local minimum solution for the current iteration. These iterations at any level are repeated until the change in parameters is below a threshold or a specified number of iterations is reached. The estimated parameters are projected to the next finer level and used as initial estimates to warp the corresponding image $I(\mathbf{p}, t-1)$, and the process repeated until convergence at the finest level. In order to generate a binary mask of regions of dominant motion and of the outliers, the residuals for the converged parameters are thresholded using a factor, typically 2.5, of the computed σ. Note that the binarization is only a post-processing step for display purposes whereas the motion computation itself uses the continuous weights derived from the ρ function.

The formulation for the 3D plane and parallax model is a bit more involved and will be presented in a later section. Presently, the use of 2D models and their results are described.

5 2D Models for Dominant Motion Estimation

We illustrate the use of the formulation in the previous section with an 8-parameter plane projective transformation. The 2D affine and translational models are specializations of the 8-parameter model. For the case of a video sequence with relatively closely spaced frames, a velocity approximation to the displacement field can be used. This approximation is valid when the rotations are small, and the change in depth of points due to motion in consecutive frames is small compared to the depth [2].

The flow field is given by $\mathbf{u}(\mathbf{p}(x,y); \boldsymbol{\theta}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix} \boldsymbol{\theta} = \mathbf{M}\boldsymbol{\theta}$. Note that $\mathbf{p}(x,y)$ refers to the actual pixel coordinates; any calibration parameters relating the pinhole-model (x,y) coordinates to the pixel coordinates have been absorbed in the unknown $\boldsymbol{\theta}$. In the iterative

direct method, this leads to a linear relationship between $\delta\mathbf{u}$ and $\delta\boldsymbol{\theta}$: $\delta\mathbf{u}(\mathbf{p};\boldsymbol{\theta}) = \mathbf{M}\,\delta\boldsymbol{\theta}$. Therefore, in each iteration the derivative of each residual of (6) w.r.t. the unknown $\boldsymbol{\theta}$ is $\frac{\partial r}{\partial \boldsymbol{\theta}} = \frac{\partial \delta\mathbf{u}}{\partial \boldsymbol{\theta}} \frac{\partial r}{\partial \delta\mathbf{u}} = \mathbf{M}^T \nabla I(\mathbf{p}, t)$. This when combined with Eqs. (4) and (5) leads to a new GN direction $\delta\boldsymbol{\theta}$ in each iteration, and subsequently a new $\boldsymbol{\theta}$ after line minimization.

5.1 Dominant Motion Separation and Mosaicing with 2D Models

In showing the results of the algorithm, we wish to emphasize that the results shown in print are nowhere near as dramatic as when shown as moving images. For all the experiments with 2D models, an initial guess corresponding to zero motion was provided to the algorithm.

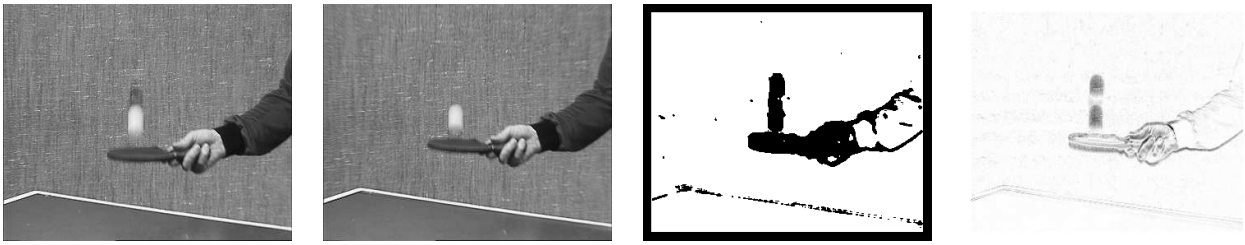

Figure 2: **Left**: Two frames, and **right**: outlier mask and difference after dominant affine warping for *tt* (Low differences are shown in white and high differences in black).

We first show the results of dominant motion separation on a sequence of table tennis play (*tt* sequence) that was obtained from a public domain archive. Two 352×288 frames of the *tt* sequence are shown in Fig. 2. The camera zooms towards the scene and the hand of the player moves up tossing the ball. The zoom is strong enough to move pixels by almost 8-10 pixels at the periphery. A 6-parameter 2D affine transformation based dominant motion algorithm was applied to this sequence. The computed motion parameters[5]: $[[1.039 \quad -0.001 \quad 0.000 \quad 1.039], [-6.397 \quad -5.816]]$, show a pure zoom factor. The background compensation, seen in a dynamic display, is almost perfect. The outlier mask and the difference image after affine warping are shown in Fig. 2. The dark areas (excepting the thin border that was automatically left out of the computation) are the outliers, and the white ones correspond to the regions of dominant motion.

The second set of results is on a sequence (*bike*) of a stunt bike rider being tracked by a camera; the sequence was obtained from a public domain archive. One 352×240 frame (No. 54) of the 88-frame sequence is shown in Fig. 3. The camera tracks the moving bike leftwards and downwards. The computed parameters, $[[0.999 \quad 0.000 \quad 0.000 \quad 1.000], [9.528 \quad -5.514]]$ show a background motion

[5]The first four parameters are for the 2×2 matrix in row major form and the other two represent the 2D translation. The origin is at the top left corner of the image.

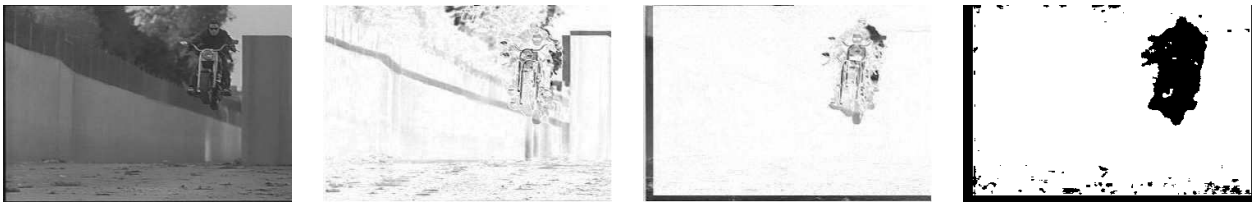

Figure 3: One frame, original difference and difference after dominant affine warping, and outlier mask for *bike*. (Low differences are shown in white and high differences in black).

of almost 10 pixels to the right and 5 pixels towards the top. The original image difference, the difference after affine warping and the outlier mask are also shown in Fig. 3.

5.2 Video Mosaicing with Dominant 2D Motion Estimation

In many real video sequences where the camera is panning and tracking an object, a panoramic view of the background can be created by mosaicing together numerous frames with warping transforms that are the result of automatic dominant motion computation. The 2D motion estimation algorithm is applied between consecutive pairs of frames. Then a reference frame is chosen, and all the frames are warped into the coordinate system of the reference frame. This process creates a mosaiced frame whose size, in general, is bigger than the original images; parts of the scene not "seen" in the reference view occupy the extra space. The mosaiced image is created by temporal filtering of the various warped images in the mosaic frame's coordinate system.

We illustrate the 2D dominant motion based mosaicing on the 88 frame *bike* video. Frames 52 and 82 are shown in Fig. 4. The dominant motion transformations are used to create a single composite frame for the whole sequence. In the dynamic version of this mosaic, we can show the stabilized (completely static) background expanding as the time proceeds, with only the bike moving. Two frames of this dynamic mosaic, again corresponding to the time instants 52 and 82 are shown in Fig. 4. Furthermore, temporal median filtering of these mosaiced frames leads to almost the deletion of the outliers (moving object in this case) if their locations are not highly correlated over time, a reasonable real scenario. Such a single frame, dominant-component-only mosaic of the *bike* sequence is shown in Fig. 4.

We wish to emphasize that our design of the 2D motion based video mosaic system can handle fairly long video sequences. In order to accomplish this, the problem of storing and manipulating long videos was addressed. In particular, from a commercial standpoint, storing videos as MPEG files was chosen since MPEG is fast becoming a de facto industry standard. The dynamic and

static mosaics are created by decompressing MPEG streams on-the-fly and storing the resulting motion and visual information. The *bike* mosaic was created with this system. In order to give a sense of the capability of the system to handle relatively long shots, Fig. 5 shows a static mosaic created automatically from a 35 second video shot (1000 frames) taken from the Yosemite Valley floor using a Hi8 hand held camcorder. Incidentally, to drive home the point that traditional video browsing on WWW is slow and cumbersome, we wish to point that when these results were put on a Web page (*http://www.almaden.ibm.com/cs/video/video_anno_ext.html*), a comment was that the original video ($25M$) is extremely slow to load but the mosaic, available as a single image, is quite fast to view.

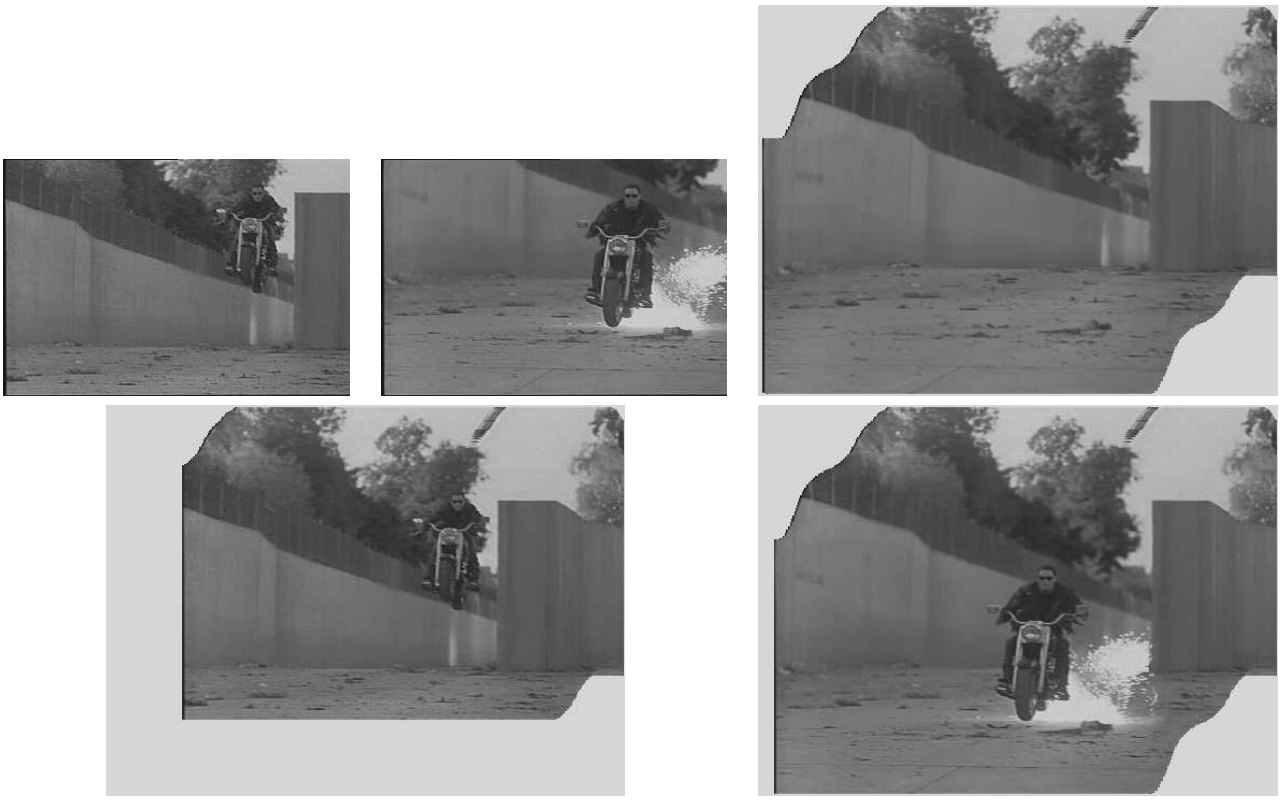

Figure 4: **Top left**: Two frames (52 and 82) of the *bike* sequence. **Top right**: Single-frame mosaic after temporal filtering. **Bottom left & right**: Dynamic mosaics at frames 52 and 82.

6 3D Model for Dominant Motion

When there are significant variations in depths of objects and surfaces in the scene, and no clear dominant 2D transformation is a good descriptor of the background motion, it is appropriate to employ models of motion and structure that account for 3D motion transformation of the view point, and for the depth of objects. We now present a model that parameterizes the transformation

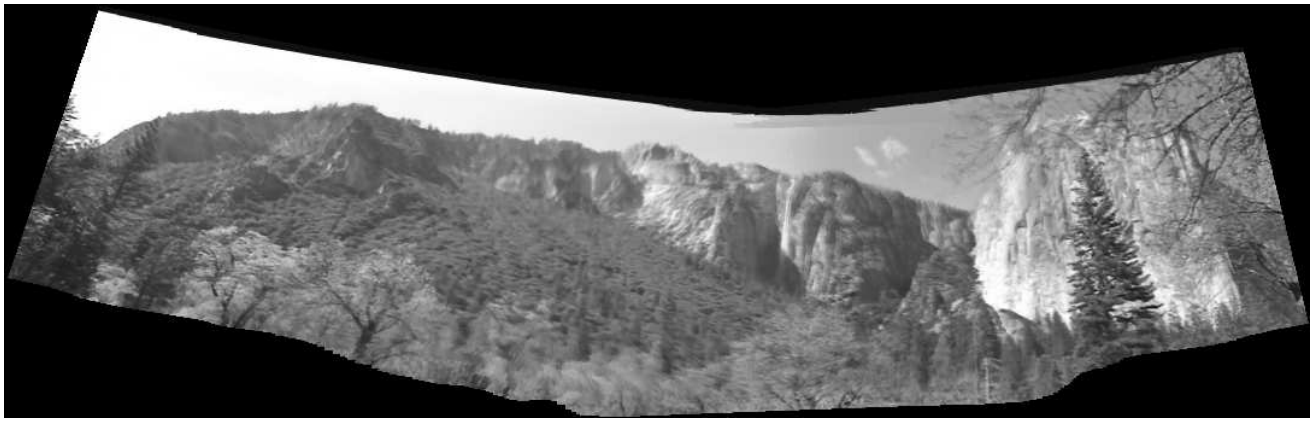

Figure 5: Single-frame mosaic of a panoramic view of Yosemite valley.

between two views of a scene in terms of a 12-parameter global projective view transformation, and a one-parameter-per-point "projective depth".

Kumar et al. [22], Sawhney [35] and Shashua and Navab [38] recently showed that given a reference view of a scene, points in any other view whose coordinate system is an arbitrary 3D affine transformation with respect to the reference view, are related to those in the reference view through: (i) a plane projective transformation for an arbitrary plane, (ii) the epipole, and (iii) the "projective depth". That is,

$$\mathbf{p}' \approx \mathbf{A}_\pi \mathbf{p} + \kappa \mathbf{t}', \tag{7}$$

where $\mathbf{p} = [x, y, 1]^T$, $\mathbf{p}'$ are the image coordinates in the reference view and another view, $\mathbf{A}_\pi$ is a 3×3 plane projective transformation corresponding to an arbitrary reference plane, $\mathbf{t}'$ is the epipole in the second image, and κ is the projective depth in the reference view. Thus, with $[\mathbf{p}^T \quad \kappa]^T$ as the homogeneous coordinates of the corresponding 3D point, $\mathbf{P}$, in the reference view, any arbitrary view of $\mathbf{P}$, $\mathbf{p}'$, is written as a view transformation, $[\mathbf{A}_\pi \quad \mathbf{t}']$, that has a planar component and an epipolar component. That is, $\mathbf{p}' \approx [\mathbf{A}_\pi \quad \mathbf{t}'][\mathbf{p}^T \quad \kappa]^T$. Therefore, $[\mathbf{p}^T \quad \kappa]^T$ is a representation of $\mathbf{P}$ that is fixed with respect to the reference view. For the sake of completeness here, if the 3D coordinates are related by $\mathbf{P}' = \mathbf{A}'\mathbf{P} + \mathbf{T}'$, and the plane is given by $\mathbf{N}^T\mathbf{P} = d$, then $\mathbf{A}_\pi \approx [\mathbf{A}' + \mathbf{T}'\mathbf{N}^T/d]$, and $\kappa = (d_N/P_z)/(-T_d/T_z)$, where d_N and T_d are the distances of $\mathbf{P}$ and $\mathbf{T}$ from the plane, respectively [22, 35, 38]. Note that the image coordinates are measured with an arbitrary coordinate system, and all the calibration parameters have been absorbed in the view transformation.

When the rotations between frames are small, and change in depth due to motion is small

compared to the depth, the image motion between two views, $\mathbf{p} = [x, y]^T$ and $\mathbf{p}'$, for a pin-hole camera model, is written as, $\mathbf{p}' = \mathbf{p} + \begin{bmatrix} -xy & 1+x^2 & -y \\ -(1+y^2) & xy & x \end{bmatrix} \mathbf{\Omega} + \frac{1}{Z} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{bmatrix} \mathbf{T}$, for a 3D rotation vector $\mathbf{\Omega}$, and 3D translation $\mathbf{T}$. Again, by choosing an arbitrary plane in the environment, the motion can be decomposed into a planar component, and a parallax component independent of rotation. Furthermore, the pin-hole model coordinates can be generalized to real pixel coordinates through scale factors (s_x, s_y), and the principal point, (c_x, c_y), potentially different for each view. Under the generally satisfied assumption in video imagery that $s_x/s_y = s'_x/s'_y$, and by an overloading of notation, the motion transformation is,

$$\mathbf{p}' = \mathbf{p} - \{\mathbf{M}(\mathbf{p})\mathbf{\Theta} + \kappa(\mathbf{p})\mathbf{B}(\mathbf{p})\mathbf{\Gamma}\} = \mathbf{p} - \mathbf{u}(\mathbf{p};\, \mathbf{\Theta}, \mathbf{\Gamma}, \kappa(\mathbf{p})), \tag{8}$$

where $\mathbf{M}(\mathbf{p})\mathbf{\Theta}$ is the small displacement approximation of a planar transformation for unknown calibration, $\mathbf{\Theta}$ is an 8-parameter vector and $\mathbf{\Gamma}$ is a 3-parameter vector representing the planar and epipole view transformation components, respectively, with the calibration parameters folded in; $\mathbf{p}$, $\mathbf{p}'$ now refer to the 2D real pixel coordinates in the two views.

6.1 Least Squares Formulation

In the *LS* formulation using the 3D model, the sum of squares function defined over the error term of (1) is used. However, since the planar transformation could correspond to *any* arbitrary plane parameters, there is an infinite family of solutions. Without increasing the complexity of the non-linear minimization problem at hand, we add a quadratic term that measures the sum of squares of the out-of-plane depths κ. The idea is to solve for the planar transformation that minimizes this sum. Thus, the following problem is to be solved,

$$\min_{\Theta, \Gamma, \kappa(\mathbf{p})} \sum_{\mathbf{p}} [I(\mathbf{p}, t) - I(\mathbf{p} - \mathbf{u}(\mathbf{p}), t-1)]^2 + \gamma \sum_{\mathbf{p}} \kappa^2(\mathbf{p}),$$

where γ is a constant. In order to solve it, we use Hanna's technique [13], which exploits the fact that $\mathbf{u}(\mathbf{p})$ is a linear and bilinear function of the unknown parameters and specializes the GN method to it. At the mth iteration in the algorithm, given estimates $\mathbf{\Theta}^{(m)}$, $\mathbf{\Gamma}^{(m)}$ and $\kappa^{(m)}(\mathbf{p})$'s, each residual is expressed as in (6), and the following error is minimized,

$$\gamma \textstyle\sum_{\mathbf{p}} \kappa^2(\mathbf{p}) + \sum_{\mathbf{p}} [\delta I(\mathbf{p}) + \nabla I^T(\mathbf{p}, t)\, \delta \mathbf{u}(\mathbf{p};\, \mathbf{\Theta}, \mathbf{\Gamma}, \kappa(\mathbf{p});\, \mathbf{\Theta}^{(m)}, \mathbf{\Gamma}^{(m)}, \kappa^{(m)}(\mathbf{p}))]^2,$$

where $\delta I(\mathbf{p}) = I(\mathbf{p}, t) - I(\mathbf{p} - \mathbf{u}(\mathbf{p}; \mathbf{\Theta}^{(m)}, \mathbf{\Gamma}^{(m)}, \kappa^{(m)}), t-1)$. Each term in the second sum is the square of a first order approximation to r in (6). Writing $\delta\mathbf{u} = \mathbf{u} - \mathbf{u}^{(m)}$, we get,

$$\min_{\Theta,\Gamma,\kappa(\mathbf{p})} \sum_{\mathbf{p}} [\delta I(\mathbf{p}) + \nabla I^T(\mathbf{p})(\mathbf{M}(\mathbf{p})(\mathbf{\Theta} - \mathbf{\Theta}^{(m)}) + \kappa(\mathbf{p})\mathbf{B}(\mathbf{p})\mathbf{\Gamma} - \kappa^{(m)}(\mathbf{p})\mathbf{B}(\mathbf{p})\mathbf{\Gamma}^{(m)})]^2 + \gamma \sum_{\mathbf{p}} \kappa^2(\mathbf{p}). \quad (9)$$

In applying the GN method, first the analytical solution of κ in terms of $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ is substituted in the above. This assumes that each $\kappa(\mathbf{p})$ within a small, $\mathcal{W} \times \mathcal{W}$, (typically 3×3) window W is constant. This gives, $\kappa(\mathbf{p}) =$

$$\{\textstyle\sum_W [\nabla I^T(\mathbf{p})\mathbf{B}(\mathbf{p})\mathbf{\Gamma}(\delta I(\mathbf{p}) + \nabla I^T(\mathbf{p})(\mathbf{M}(\mathbf{p})(\mathbf{\Theta} - \mathbf{\Theta}^{(m)}) - \kappa^{(m)}(\mathbf{p})\mathbf{B}(\mathbf{p})\mathbf{\Gamma}^{(m)})]\} / \{\textstyle\sum_W [\nabla I^T(\mathbf{p})\,\mathbf{B}(\mathbf{p})\,\mathbf{\Gamma}]^2 + \gamma\mathcal{W}^2\}.$$

This expression is substituted in (9) and the GN method applied to solve for $\delta\mathbf{\Theta}$ and $\delta\mathbf{\Gamma}$. Given the new estimates of these view parameters, the new κ's are solved for numerically now using the above equation. Again, the proper step sizes for $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ are chosen using line search. The iterations are repeated over multiple resolutions. The initial parameters chosen for the 3D estimation are: $[0\ 1\ 0\ 0\ 0\ 1\ 0\ 0]^T$ for $\mathbf{\Theta}$, i.e. zero motion for the planar part, all zeros for the κ's, and $[1\ 0\ 0]^T$ for the $\mathbf{\Gamma}$.

6.2 Image Registration and Mosaicing using 3D Parameters

First, two images are registered while solving for the view and depth parameters with respect to one as a reference view. Subsequently, to verify the goodness of depth computation, *only* the view parameters, $\mathbf{\Theta}$ and $\mathbf{\Gamma}$, are solved for between a third view and the reference view with the κ's kept fixed (see (7)) from the first computation.

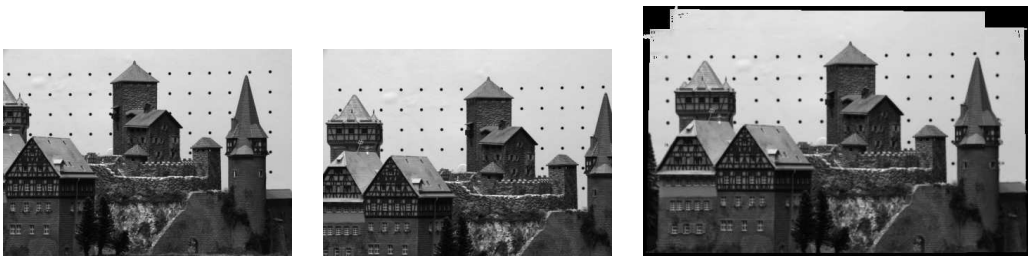

Figure 6: **Left**: Two frames (1 and 6) of *castle*. **Right**: Single-frame mosaic after temporal filtering.

Four frames (1, 3, 4 and 6) of a 6-frame *castle* sequence are shown in Fig. 6. The range of depth is between 40 to $300mm$. Frames 1–4 were obtained by a sideways motion of the camera, and frames 5–6 by an upwards motion. There is about 15–30 pixels motion between consecutive frames.

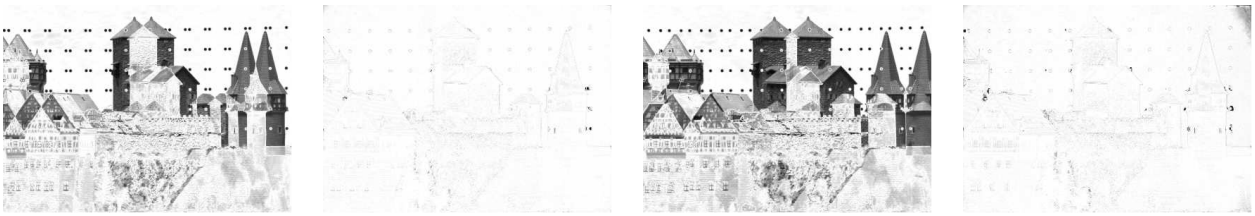

Figure 7: **Left two**: Original difference (left), and difference (right) between warped (with plane+parallax) frame 2 and frame 1 for *castle*. **Right two**: Original difference (left), and difference (right) between warped (with plane+parallax) frame 3 and frame 1. (Low differences are shown in white and high differences in black).

First, with frame 1 as reference, the planar and parallax transformation (Eq. (8)) are solved for between frames 1 and 2. Fig. 7 shows the difference image for the original frames 1 and 2, and that for the difference between frame 1 and frame 2 warped using the computed 3D view transformation and depth. In order to further test the effectiveness of the computed depth representation with respect to the reference frame, *only* the view parameters are solved for between frame 1 and a new frame 3, with the depths kept constant as computed previously. The result of this process is shown in Fig. 7; the first image is the original difference and the second is the difference after warping the third image towards the reference using the newly computed view parameters, and the reference view centered depths computed earlier.

For registration, the warped image, $I^w(\mathbf{p})$ is created in the coordinate system of the reference image, by transforming the second image, say I, using $I^w(\mathbf{p}) = I(\mathbf{p} - \mathbf{u}(\mathbf{p}; \mathbf{\Theta}, \mathbf{\Gamma}, \kappa(\mathbf{p})))$, where $\mathbf{u}$ is as in (8), and $\mathbf{p}$ is in the reference image's coordinate system. Both the view transformation and depth computations are reasonable since all three frames are registered reasonably well.

Now the result of mosaicing frames 1–6 in the coordinate system of frame 3 is shown. First, view parameters and depth between frames 1–2, 4–3 and 6–5 are solved with frames 1, 4 and 6 as the reference. Subsequently, with the depth parameters from the previous step, only the view parameters were solved between frames 1–3 and 6–3, again with 1 and 6 as reference. Finally, frames 1, 4 and 6 were forward warped using these view parameters and the depths, and mosaiced with frame 3 in 3's coordinate system. The outcome is shown in Fig. 6. Note that the steeple castle on the right, and the house next to it, that are almost not seen in view 3, are fully there in the mosaiced frame. Also, the house on the left is completely visible. The size of the original frames is 411×295 and that of the mosaic is 501×326.

We have compared the results of mosaic creation with only a 2D affine transformation in this

case. Due to lack of space, the reader is referred to [4, 5]. The parallaxes that the affine transform is unable to account for are evident through non-zero differences and the blurry mosaic. The uncompensated parallax is dramatically evident when a motion display of warped and the reference images is shown.

We have extended the 3D estimation algorithm to the case of multiple moving objects using a robust formulation akin to the 2D case. This algorithm can estimate the 3D parameters of the fixed scene and separate moving objects as outliers. However, due to lack of space we cannot present the work here and refer the reader to [4, 36] for details.

7 Discussion on Dominant vs. Multiple Motion Estimation

It is clear from the experimental demonstrations using dominant motion estimation that for many realistic scenarios, where a single 2D model captures the motion of the dominant background reasonably well, compact representations of videos can be created with dominant 2D models. Furthermore, successive application of dominant motion estimation on a video may be used to capture scene and object motions with 2D models. When 2D models do not suffice, 3D models may be employed albeit at the expense of keeping dense depth representations around. The visual representation of a complete video shot as a panoramic mosaic image enables browsing and indexing into arbitrarily selected portions of the shot by using the mosaic as a window into the contents of the shot. Dynamic objects and events can be represented separately using their own visual appearance and motion representations. The dominant motion representation tends to group together large parts of the static scene into single regions with a single motion model. However, for indexing videos based on objects and significant structures in static scenes as well as on moving objects, a finer motion analysis that leads to segmentation of multiple layers of motion and structure may be required. A formulation of this problem and an algorithm is the subject of the next sections.

8 Simultaneous Estimation of Multiple Motions

8.1 Mixture Models for Maximum-Likelihood Estimation

We formulate the problem of multiple motions and layer estimation as the optimization of an objective function, whose minimum is expected to lead to the best description of an image and of its change over time as described by the parameters of the motion models, the proportions of these models in the data, and the ownership weights (the layers) for each model.

Given a pair of images captured at time instants $t-1$ and t in a sequence, the image at t, the

reference image, is modeled as being generated by that at $t-1$ through a finite mixture of warped images, each being warped using its own motion model. The intensity $I(\mathbf{p}, t)$ at pixel $\mathbf{p}$ and time t is modeled as arising from a superpopulation of intensity maps, $\tilde{\mathbf{I}}$. $\tilde{\mathbf{I}}$ is a set of g maps $\{\tilde{I}_1, \ldots, \tilde{I}_g\}$, where $\tilde{I}_i$ represents the predicted image at time t as a function of the image at time $t-1$ and of the ith motion model parameters $\boldsymbol{\theta}_i$, that is,

$$\tilde{I}_i(\mathbf{p},\, \boldsymbol{\theta}_i,\, t) = I(\mathbf{p} - \mathbf{u}(\mathbf{p},\, \boldsymbol{\theta}_i),\, t-1). \tag{10}$$

The probability density function (pdf) of the reference intensity map, $I(\mathbf{p}, t)$, as predicted from $\tilde{I}$ can be represented in the finite mixture form [28] as

$$f(I(\mathbf{p},\, t) \mid I(\mathbf{p},\, t-1), \boldsymbol{\Phi}) = \sum_{i=1}^{g} \pi_i\, p_i(I(\mathbf{p},\, t) \mid \tilde{I}_i(\mathbf{p},\, \boldsymbol{\theta}_i,\, t), \sigma_i). \tag{11}$$

In the following, it will be assumed that $I(\mathbf{p}, t-1)$ is given and thus not included explicitly in the notations. Also, unless necessary, $I(\mathbf{p}, t)$ and $\tilde{I}_i(\mathbf{p},\, \boldsymbol{\theta}_i,\, t)$ are written as $I(\mathbf{p})$ and $\tilde{I}_i(\mathbf{p})$, respectively. Note that this formulation does not account for the presence of observations that are not modeled by the constancy of brightness under modeled motions, that is for outliers of the complete mixture of models. However, instead of introducing an additional component in the mixture meant to model these outliers as in [21, 26], we solve this problem through a combination of a rejection procedure and of robust estimation of the mixture parameters. This will be explained in Section 8.4. Also note that the formulation in this section is for an arbitrary but *fixed* g. The problem of estimating an optimal g is the subject of Section 9.

In the finite mixture formulation, the vector $\boldsymbol{\Phi}$ represents the vector of all unknown parameters $\boldsymbol{\Phi} = \left[\boldsymbol{\Pi}^T,\, \boldsymbol{\Sigma}^T,\, \boldsymbol{\Theta}^T\right]^T$, with $\boldsymbol{\Pi} = [\pi_1, \ldots, \pi_g]^T$, $\boldsymbol{\Sigma} = [\sigma_1, \ldots, \sigma_g]^T$, and $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_g]^T$. In this notation, $\boldsymbol{\theta}_i$ denotes the parameter vector, σ_i^2 the variance, and π_i the proportion of the ith model in the mixture such that

$$\sum_{i=1}^{g} \pi_i = 1, \quad \pi_i > 0, \quad i = 1, \ldots, g. \tag{12}$$

We assume that each $p_i(I(\mathbf{p}) \mid \tilde{I}_i(\mathbf{p}), \sigma_i)$ belongs to the same parametric family, p. In particular, $p(I(\mathbf{p}) \mid \tilde{I}_i(\mathbf{p}), \sigma_i)$ is assumed to be a normal distribution, $\mathcal{N}(\tilde{I}_i(\mathbf{p}), \sigma_i^2)$, with spatially varying mean $\tilde{I}_i(\mathbf{p})$ and variance σ_i^2. Under the assumption that the N observations $I(\mathbf{p})$, one at each pixel,

are realized values of N independent and identically distributed random variables with common distribution function $p(I(\mathbf{p}) \mid \mathbf{\Phi})$, the negative log-likelihood function $L(\mathbf{\Phi})$ of the parameters $\mathbf{\Phi}$, given the observations, can be written as

$$L(\mathbf{\Phi} \mid \{I(\mathbf{p}_j)\}) = -\log(\prod_{j=1}^{N} f(I(\mathbf{p}_j) \mid \mathbf{\Phi})) = -\sum_{j=1}^{N} \log\left(\sum_{i=1}^{g} \pi_i \; p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i)\right), \tag{13}$$

where $\{I(\mathbf{p}_j)\}$ denotes the N observations $I(\mathbf{p}_1), \ldots, I(\mathbf{p}_N)$.

Given estimates of $\mathbf{\Phi}$, the estimates of the posterior probabilities of population membership can be formed for each observation, $I(\mathbf{p}_j)$, to generate weights for each layer. The estimate of the ownership weight at the jth pixel location for the ith population, $\tilde{I}_i$, is given by

$$\begin{aligned} \tau_{ij} &= prob(\mathbf{p}_j \in \tilde{I}_i \mid I(\mathbf{p}_j); \mathbf{\Phi}) = \frac{prob(\mathbf{p}_j \in \tilde{I}_i \mid \mathbf{\Phi}) \; p(I(\mathbf{p}_j) \mid \mathbf{p}_j \in \tilde{I}_i; \mathbf{\Phi})}{f(I(\mathbf{p}_j) \; ; \mathbf{\Phi})} \\ &= \frac{\pi_i \; p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i)}{\sum_{t=1}^{g} \pi_t \; p(I(\mathbf{p}_j) \mid \tilde{I}_t(\mathbf{p}_j), \sigma_t)} \end{aligned} \tag{14}$$

It can be shown ([16, 28]) that the maximum likelihood estimates of $\mathbf{\Phi}$, $\hat{\mathbf{\Phi}}$, satisfy

$$\hat{\pi}_i = \sum_{j=1}^{N} \frac{\hat{\tau}_{ij}}{N}, \qquad i = 1, \ldots, g, \tag{15}$$

$$\sum_{i=1}^{g} \sum_{j=1}^{N} \hat{\tau}_{ij} \frac{\partial \log(p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i))}{\partial \hat{\sigma}_i} = 0, \tag{16}$$

$$\sum_{i=1}^{g} \sum_{j=1}^{N} \hat{\tau}_{ij} \frac{\partial \log(p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i))}{\partial \hat{\boldsymbol{\theta}}_i} = 0. \tag{17}$$

8.2 Specialization to Binary Ownerships

In accordance with the formulation in [28, pg. 14], the negative log-likelihood function given in (13) is now presented for the case when each measurement is mutually exclusively associated with a single model. For this purpose, for each measurement j, a g-dimensional vector of indicator variables $\mathbf{z}_j = [z_{1j}, \ldots, z_{gj}]^T$ is introduced, where

$$z_{ij} = \begin{cases} 1, & I(\mathbf{p}_j) \in \tilde{I}_i, \\ 0, & I(\mathbf{p}_j) \notin \tilde{I}_i, \end{cases} \tag{18}$$

and $\mathbf{z}_1, \ldots \mathbf{z}_N$ are independent and identically distributed according to a multinomial distribution consisting of one draw on g categories with probabilities $\pi_1, \ldots \pi_g$. Therefore, the pdf $f(I(\mathbf{p}_j) \mid \mathbf{\Phi},\ \mathbf{z})$, is given by

$$f(I(\mathbf{p}_j) \mid \mathbf{\Phi},\ \mathbf{z}) = \prod_{i=1}^{g} [\pi_i\, p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i)]^{z_{ij}}. \tag{19}$$

Again under the assumption of measurements being independent, the complete negative log-likelihood function for all the measurements, conditioned on the indicator variables, is given by

$$L_C(\mathbf{\Phi} \mid \{I(\mathbf{p}_j)\};\ \mathbf{z}_1, \ldots, \mathbf{z}_N) = -\sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\, \{\log(\pi_i) + \log(p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i))\} \tag{20}$$

In the process of estimation, the indicator variables are assigned as $z_{ij} = 1$ if the ownership probability

$$\tau_{ij} > \tau_{tj}, \quad t = 1, \ldots, g;\ t \neq i,$$

and 0 otherwise.

8.3 Iterative Solution for ML Estimation of Mixture Parameters

Eqs. (15)–(17) and (14) suggest an iterative solution for the maximum likelihood estimates of $\mathbf{\Pi}$, $\mathbf{\Sigma}$ and $\mathbf{\Theta}$, and for the posterior ownership probabilities, τ's. One such solution method is called the Expectation-Maximization (EM) algorithm, that proceeds iteratively in two steps, E and M. The M-step solves for the ML estimates of the mixture model parameters, and the E-step for the ownership probabilities. For the particular case of a mixture of normal distributions, the M- and E-step of the classical EM algorithm may be expressed in a closed-form solution. However, in our case, the ML estimates of the motion parameters $\mathbf{\Theta}$ have no closed-form solution. Instead, our algorithm implements the ML estimation of the motion parameters using the dominant motion algorithm. First, $\mathbf{\Theta}$ is solved for given values of $\mathbf{\Pi}$ and $\mathbf{\Sigma}$ and then the optimal estimates for $\mathbf{\Pi}$ and $\mathbf{\Sigma}$ are computed using the estimated motion parameters. A Newton iterative scheme including all parameters into a single estimation process would be an alternative to solve the problem.

8.4 Estimation of the Mixture Parameters

8.4.1 Likelihood Estimation for a Mixture of Normal Distributions

With the component densities of the mixture assumed to be normal, the likelihood equations given in (15) and (16) can be used for an iterative computation of the solution by the EM algorithm [28,

pg. 38]. Given estimates of the parameters $\mathbf{\Theta}$, $\mathbf{\Sigma}$, and $\mathbf{\Pi}$, in the E-step, the posterior probability that the jth pixel belongs to the ith model, i.e. τ_{ij}, is computed as

$$\tau_{ij} = prob(\mathbf{p}_j \in \tilde{I}_i \mid I(\mathbf{p}_j); \mathbf{\Phi}) = \frac{\pi_i \; p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i)}{\sum_{t=1}^{g} \pi_t \; p(I(\mathbf{p}_j) \mid \tilde{I}_t(\mathbf{p}_j), \sigma_t)} = \frac{\frac{\pi_i}{\sigma_i} \exp\left(-\frac{r_i(\mathbf{p}_j)^2}{2\sigma_i^2}\right)}{\sum_{t=1}^{g} \frac{\pi_t}{\sigma_t} \exp\left(-\frac{r_i(\mathbf{p}_j)^2}{2\sigma_t^2}\right)}, \qquad (21)$$

where $r_i(\mathbf{p}_j) = I(\mathbf{p}_j, t) - \tilde{I}_i(\mathbf{p}_j, \boldsymbol{\theta}_i, t)$.

The M-step consists of solving the likelihood equations (15), (16) and (17) with each $\hat{\tau}_{ij}$ replaced by its value computed in the E-step. Eq. (15) already presents the solution for $\hat{\pi}_i$. The step for the ML estimate of the motion parameters, $\mathbf{\Theta}$, will be detailed in Section 8.5. For the case of Gaussian distributions, the solution for the σ's is given by: $\hat{\sigma}_i^2 = \sum_{j=1}^{N} \frac{\hat{\tau}_{ij} \; r^2(\mathbf{p}_j)}{N \; \hat{\pi}_i}$.

The E- and M-steps are repeated alternately, where in their subsequent executions, the initial fit $(\mathbf{\Pi}, \mathbf{\Sigma})^{(m)}$ of the parameters is replaced by the current fit $(\mathbf{\Pi}, \mathbf{\Sigma})^{(m+1)}$.

8.4.2 Robust Scale Estimation

An alternative to the weighted squared residual estimation of σ above is to use a robust estimate. It is to be emphasized that a good estimation of σ is critical to the estimation of both the motion parameters and layers of ownership weights. In the M-step, a robust estimate of σ for each of the binary layers is computed using the median value of the residuals for the layer in accordance with the scale estimation for the dominant motion explained in Section 4.2. In other words, each set of z_{ij}'s for a given model i provide a set of pixels j over which the robust scale is estimated.

8.4.3 Detection of Atypical Observations

Outliers, within the context of the mixture model, can arise essentially due to the violation of the brightness constancy constraint employed in the direct method for motion estimation (see Sections 4 and 8.5). This constraint is violated when (i) brightness patterns do not move according to the coordinate transform specified by the motion parameters, for instance, motion of highlights and light sources, and (ii) when due to occlusion/deocclusion intensity patterns visible in one image are missing in the other.

Atypical observations must be detected and removed from the data, so that they do not influence the estimation of the mixture parameters any more. One way of evaluating whether an observation is from a mixture of intensity maps $\tilde{I}_1, \ldots, \tilde{I}_g$ is to assess how typical this observation is of each $\tilde{I}_i$ taken in turn. An observation which is atypical of each $\tilde{I}_i$ may well be considered as a contaminant which does not belong to the mixture. In [28, pg. 62], it is shown that, for sufficiently large N, the typicality of an observation can be approximated by the area to the right of the Mahalanobis distance of the measurement with greatest posterior probability, $\hat{\tau}_{ij}$, under the χ^2_p distribution, where p is the dimensionality of the Gaussian distributions. Thus, in our one-dimensional case, the typicality of an observation $I(\mathbf{p}_j)$ can be approximated by the area to the right of $\sum_{i=1}^{g} \hat{z}_{ij} \frac{r_i^2(\mathbf{p}_j)}{\hat{\sigma}_i^2}$, under the χ^2_1 distribution, and where $\hat{z}_{ij}$ is as in (18). In turn, this test coincides with a simple test of the form $\min_i \frac{r_i^2(\mathbf{p}_j)}{\hat{\sigma}_i^2} > c^2$. In our experiments, we choose $c = 2.5$.

8.5 Estimation of the Motion Parameters

In the iterative solution of the mixture model parameters introduced in Section 8, one of the M-steps consists of computing the new estimates of the motion parameters $\boldsymbol{\Theta}$, given the current estimates of the variances $\boldsymbol{\Sigma}$, mixture proportions $\boldsymbol{\Pi}$, ownership weights τ's ($\mathbf{z}$'s), and motion parameters. This corresponds to a solution of the necessary condition for the ML estimate of $\boldsymbol{\Theta}$ given in (17), which unfortunately cannot be expressed in a closed form. Maximizing the likelihood function is equivalent to minimizing the negative of its logarithm (the log-likelihood), which in turn is equivalent to minimizing $\sum_{j=1}^{N} h(\boldsymbol{\theta_i}) = \sum_{j=1}^{N} \tau_{ij}\, \rho(r_i(\mathbf{p}_j); \sigma_i)$ where the function ρ is related to the likelihood function for an appropriate choice of the error distribution.

As explained in the preceding section, the probability distributions of the residuals given the parameters are modeled as a mixture of contaminated Gaussians. Thus, in order to allow for outlying data, instead of least squares estimation of the parameters, we use robust M-estimators, and compute the motion parameters by using the dominant motion method of Section 4.

9 Multiple Motion Formulation Using MDL Encoding

9.1 MDL Formulation for Determining Model Complexity

The problem with the maximum likelihood formulation of Eqs. (15)–(17) is that there is no bound on the complexity of the mixture model, i.e. on the number of populations g. Generally,

the more the number of models, the better the obtained fit will be. We address this problem by applying the Minimum Description Length (MDL) principle. The first reason for choosing MDL is its information-theoretic grounding: the model that can be encoded the cheapest while explaining the observations is the best. For this purpose, the number of bits required to encode the model and the residuals is used. The goal is then to find the model parameters $\mathbf{\Phi}$ that minimize the total encoding length. A second important reason is that the MDL principle leads to an objective function with no arbitrary thresholds. For the sake of clarity, we present the MDL formulation only for the binary case and leave a detailed comparison of the binary and non-binary formulations to a subsequent paper.

The encoding has two parts, one part for the model and the other for the data using the model. The overall codelength to be minimized is

$$\mathcal{L}(\{I(\mathbf{p}_j)\}, \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) + \mathcal{L}_D(\{I(\mathbf{p}_j)\}) \mid \mathbf{\Phi}), \tag{22}$$

where $\mathcal{L}$, $\mathcal{L}_M$ and $\mathcal{L}_D$ denote the appropriate encoding length in terms of bits for the corresponding entities to be encoded.

The model parameters consist of three different components. Thus, $\mathcal{L}_M(\mathbf{\Phi}) = \mathcal{L}_{M_1}(\mathbf{\Pi}) + \mathcal{L}_{M_2}(\mathbf{\Sigma}) + \mathcal{L}_{M_3}(\mathbf{\Theta})$. For computing the coding cost of these real-valued parameters, the expression derived by Rissanen [33] in his optimal precision analysis is used. For encoding K independent real-valued parameters characterizing a distribution used to describe/encode N data points, the codelength is $(K/2)\ \log(N)$. Rissanen derives this expression for the encoding cost of real-valued parameters by optimizing the precision to which they are encoded. Thus, $\mathcal{L}_M(\mathbf{\Phi}) = \frac{K}{2}\log(N)$ where K is the total number of parameters and N is the number of pixels in the image.

Furthermore, we need to encode the data given the model $\mathcal{L}_D(\{I(\mathbf{p}_j)\}) \mid \mathbf{\Phi})$. Since we know the probability, $P(I(\mathbf{p}) \mid \mathbf{\Phi})$, from the mixture model, the optimal number of bits required to encode this is just the negative logarithm of the probability [33]. Therefore, this term is directly derived from the negative log-likelihood of the data given the model, presented in (20), by replacing the pdf, $p(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i)$ by the corresponding probability, $P(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i)$. Under the assumption of normal distribution of the residuals, and if the residuals are quantized to the nearest ϵ, their real precision, then [24]

$$P(I(\mathbf{p}_j) \mid \tilde{I}_i(\mathbf{p}_j), \sigma_i) \approx \frac{\epsilon}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-r_i^2(\mathbf{p}_j)}{2\sigma_i^2}\right), \quad \text{when } \epsilon < \sigma_i.$$

Therefore, by substituting this in (20), the total encoding length is given by

$$\mathcal{L}_B(\{I(\mathbf{p}_j)\}), \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) - \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\left\{\log(\pi_i) + \log\left(\frac{\epsilon}{\sqrt{2\pi}\sigma_i}\exp\left(\frac{-r_i^2(\mathbf{p}_j)}{2\sigma_i^2}\right)\right)\right\}.$$

By eliminating the terms independent of g, these can be simplified to

$$\mathcal{L}_B(\{I(\mathbf{p}_j)\}), \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) + \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\left\{-\log(\pi_i) + \log(\sigma_i) + \frac{r_i^2(\mathbf{p}_j)}{2\sigma_i^2}\right\} \tag{23}$$

9.2 MDL Formulation with Statistical Dependencies

We now give an MDL formulation that takes into account the statistical dependencies between the layer ownerships of neighboring pixels for each motion model. Eq. (23) can be rewritten as

$$\mathcal{L}_M(\mathbf{\Phi}) + \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\left\{\log(\sigma_i) + \frac{r_i^2(\mathbf{p}_j)}{2\sigma_i^2}\right\} - \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\log(\pi_i),$$

where the last term represents the encoding length associated with the representation of the layers of support of each model, i.e. z_{ij}, using a zero-order statistical model.

The information contained in the $\mathbf{z}$'s is characterized by a high level of statistical dependencies, or correlation, which are not taken into account in the above formulation, thus leading to an exaggerated measure of its real information content. This high level of correlation is due to the fact that pixels belonging to the same motion generally form compact regions. We capture the statistical dependencies of z_{ij} by performing a first-order linear prediction on the values of z_{ij} and by encoding only the innovation process, i.e. the prediction errors. Based on this, Eq. (23) can be transformed to

$$\mathcal{L}_{PB}(\{I(\mathbf{p}_j)\}), \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) + \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\left\{\log(\sigma_i) + \frac{r_i^2(\mathbf{p}_j)}{2\sigma_i^2}\right\} + \mathcal{L}(\mathbf{Z}), \tag{24}$$

where $\mathcal{L}(\mathbf{Z})$ represents the encoding length associated with the z_{ij} following a first-order linear prediction scheme.

The need for such a formulation can be explained as follows. It is well known that the use of a zero-order statistical model like in (23) will always yield a decrease of the encoding length when the alphabet size is decreased, i.e. when a model is suppressed. This decrease will in general

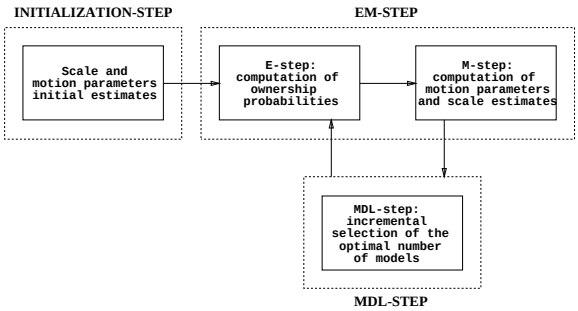

Figure 8: Flow chart of the algorithm

coincide with an increase of the residual encoding length. However, the presence of noise in the data can lead to the erroneous suppression of a model if the information contained in the $\mathbf{z}$'s is overestimated. In this case, the increase of the residual encoding length is partly masked by the noise contribution, and thus may not balance the decrease in the label encoding length.

10 Minimization Procedure for Estimating g and the Mixture Parameters

Eqs. (23) and (24) are expressions for the complete encoding lengths of the models and the data given the models. Ideally, optimization of these encoding lengths with respect to *all* the unknowns should be performed. However, this will obviously be prohibitively expensive given the enormous parameter space. In order to circumvent this practically impossible task, we divide the problem into alternating steps of ML estimation of the mixture parameters, detection of outliers given these mixture parameters, and pruning of the number of models using the encoding criterion. We use a descending procedure which, given a set of motion and mixture parameters, computes the encoding length by removing a single model from the population. If the encoding length decreases for at least one of those, in comparison with the encoding length computed for the full set of models, then the model with the largest decrease in the codelength is removed from the population. Fig. 8 shows a flow chart of the algorithm. The algorithm may be decomposed into three different parts: the initialization step, the EM step, and the MDL step.

In the initialization step, initial estimates of the motion parameters and the σ's are generated. Rectangular tiled binary layers that cover the entire image are defined. The number of these tiles is a user defined parameter. Typically, 16 tiles are used by dividing the x and y dimensions into four equal parts each. Thus, 16 non-overlapping binary masks are used to compute 16 initial motion parameters. In each of the subregions, motion parameters and σ's are computed independently only at the coarsest level of the pyramid using the robust estimation technique described in Section 4.

After the initialization step, the initial motion parameters and scale estimates become the current estimates used in the EM-step. For all the experiments, 2D 6-parameter affine models have been used as the motion descriptor for each layer.

The next part of the algorithm is the EM-step, which again may be decomposed into two parts: the E-step and the M-step. Given the current estimates of the motion parameters, of the σ's, of the π's, and of the number of models g, the ownership weights (τ's and $\mathbf{z}$'s) are computed for the support layers over the complete image. This is the E-step. The M-step consists of the computation of the new σ's, of the new model proportions π's, and of the new motion parameters using the new support layers. Note that each M-step employs the dominant motion algorithm (Section 4.2) for each of the g motion models. After an EM-step, outliers are detected and removed for the next MDL and EM steps.

The next part of the algorithm is the MDL-step. Following the EM-step, the total encoding length and the encoding lengths that would result by removing in turn a single layer are computed. The layer and the motion model which leads to the largest decrease, if any, is eliminated. A new EM-step is then performed with the new number of models, again followed by a MDL-step. The whole process is repeated at a given level until both the motion parameters and the number of models have converged. The motion parameter, scale and layer estimates obtained at this level are finally projected down and the same process is repeated at the next finer level.

A Maximum A-Posteriori (MAP) estimation may be optionally applied to the layer ownerships obtained once the EM-MDL steps have converged to a solution. This leads to further grouping of small, isolated layers into bigger and homogeneous layers. However, for the purposes of computing features for appearance-based indexing, this step may be unnecessary. We have implemented this step too but are unable to present further details due to lack of space (see [4] for details).

11 Results of Simultaneous Multiple Motion Estimation

We show the results of layered motion estimation using binary weights. It is to be demonstrated that the proposed scheme is robust in the presence of moving objects and is also general enough to deal with scenes with moving or static cameras. For the different sequences, one frame from the original sequence is shown. Also shown are the labeled layers and the outliers along with the residual errors between the reference image and the mosaiced image created by warping the reference image using each of the motion models for each layer. Pixel locations which correspond

to outliers have been assigned a zero value in the mosaiced frame, and the value corresponding to the non-compensated image difference at this pixel location in the mosaiced difference. The binary layers are shown both after the EM-MDL estimation, as individual regions in white, and also as a composite labeled image after EM-MDL-MAP estimation with each layer in a different shade of grey and the outlier layer in black.

It is important to note that, for some sequences (like the box or flower garden sequence), there is no clear dominant motion except maybe for the background motion. In these cases, sequential estimation of dominant motion ends up finding some average motion parameters for some layers. However, our simultaneous competitive method leads to meaningful layer descriptions because measurements are allowed to choose the best model amongst a few.

The first image sequence captures a situation, where the scene is static but the camera motion induces parallax motion onto the image plane due to the different depths in the scene. The sequence is the well known flower garden sequence. Fig. 9 shows one original image, the result for the binary layers, the outlier mask, and the residual error between the original images and that between the reference and warped mosaiced image. Note that the occlusion region at the right edge of the tree is well captured as outliers. Fig. 10 shows each of the layers.

The next results are shown on a sequence of a box rotating around its vertical axis. In this scene, the camera is static, and hence so is the background in the images. Fig. 11 shows one original image, the layers and outliers, and the residual error between the original images and that between the reference and warped mosaiced image. Fig. 12 shows each of the layers.

To illustrate the situation where both the camera and the objects are moving, results on the table tennis sequence are shown. The result of our algorithm is two layers representing the background and the hand, with good compensation for each of them. Results for this sequence are given in Figs. 13 and 14. The information about the motion of the ball is contained only at its boundary which has very small support. Therefore, the ball has been detected in the outlier layer. Also, there is a jitter/noise at the edges of the table in the original video. These unmodeled measurements have also been grouped into the outlier layer.

12 Conclusions

Compact but visually authentic video representation is increasingly becoming an important issue in the context of video indexing and annotation, and low bit rate video encoding. We have focused

on the use of motion analysis to create visual representations of videos that may be useful for efficient browsing and indexing in contrast with traditional frame oriented representations. Two major approaches for motion based representations have been presented. The first approach demonstrated that dominant 2D and 3D motion techniques are useful in their own right for computing video mosaics through the computation of dominant scene motion and/or structure. However, this may not be adequate if object level indexing and manipulation is to be accomplished efficiently. The second approach that we presented addresses this issue through simultaneous estimation of an adequate number of simple 2D motion models. A unified view of the two approaches naturally follows from the multiple model approach: the dominant motion method becomes a particular case of the multiple motion method if the number of models is fixed to be one and only the robust EM algorithm without the MDL stage is employed.

There are tradeoffs between the two major approaches to computing motion based descriptions of videos. The simplicity of formulation and the associated algorithm for a sequential dominant motion approach has been exploited by a number of researchers. However, the fact that image measurements are not able to compete for the ownership of different models leads to the use of externally specified thresholds in deciding the model to pixel association for a layer. Simultaneous estimation of the motion parameters and their layers of support adds to the complexity of the formulation and the algorithm. In general, the number of layers, their motion parameters and the support weights are to be found simultaneously. We have found that the formalism of mixture models and MDL encoding is one systematic way of capturing all the unknowns in a single optimization problem. None of the sequential dominant motion approaches in the literature have shown promising results for accurate multiple motions and layer estimation as our algorithm does.

A number of issues have been dealt with only summarily or have been left out. These issues can be broadly divided into two sets. The first set deals with issues related to the specific techniques presented in this paper and the related work. The second set involves issues of representation of scenes and moving objects for indexing and querying. In the context of the techniques presented in this paper, first, inclusion of 3D models in the representation with layers has not been experimented with. Second, extension of the layered motion algorithm to multiple frames is currently being explored. A simple extension is to use the layers computed from two frames as initial guesses for subsequent frames. However, questions like when to instantiate a new layer and when to abandon an old one need to be dealt with. Video mosaics can be easily extended for the case

of multiple layers. A composite mosaic can be created by combining the mosaics of each layer using the respective parametric motion models. However, from the viewpoint of representations, how to represent moving objects, especially articulated and non-rigid objects, compactly is still an open research issue. Combining 2D and 3D layered representations of the static scene and moving objects for video sequences addresses the problems of model-based compression, and of compact visual representations that will aid the process of visual appearance based indexing and annotation. Creating stable representations of appearances for search and recognition of objects and scenes through indexing is an important area that has not been the focus of this paper.

We are actively investigating the usefulness of motion based video representations in the context of video indexing and annotation. With the constant increase in the processing power of workstations and desktops, and the common availability and use of images and videos on these, computer vision algorithms for intermediate representations may indeed become viable and useful to intelligently manage the enormous amounts of data at hand.

Acknowledgments

We wish to thank Monika Gorkani for her help in implementation of the dominant motion mosaic code, and for locating interesting MPEG sequences. Valuable comments from the reviewers helped in focusing the paper on the key themes. We would also like to thank Dragutin Petkovic, Wayne Niblack, Byron Dom and M. Flickner for all their support during the course of this work.

References

[1] E. H. Adelson and P. Anandan. Ordinal characteristics of transparency. In *Proc. AAAI Workshop on Qualitative Vision*, 1990.

[2] G. Adiv. Determining 3D motion and structure from optical flows generated by several moving objects. *IEEE PAMI*, 7(4):384–401, 1985.

[3] J. Ashley, M. Flickner, J. Hafner, et al. Automatic and semi-automatic methods for image annotation and retrieval in QBIC. In *Image and Video Storage and Retrieval III*, volume 2420, San Jose, CA, 1995. SPIE.

[4] S. Ayer. *Sequential and Competitive Methods for the Estimation of Multiple Motions.* PhD thesis, EPFL, Lausanne, 1995.

[5] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proc. Intl. Conf. on Computer Vision*, pages 777–784, 1995. ftp://eagle.almaden.ibm.com/pub/cs/reports/vision/layered_motion.ps.Z.

[6] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *ECCV*, Stockholm, Sweden, May 1994.

[7] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model–based motion estimation. In *2nd ECCV*, pages 237–252, 1992.

[8] M. J. Black and P. Anandan. The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. Technical Report TR, Xerox PARC, CA, Dec. 1993.

[9] M. Bober and J. Kittler. Robust motion analysis. In *CVPR*, pages 947–952, Seattle, USA, June 1994.

[10] T. Darrell and A. P. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, 1995.

[11] M. Flickner, H. S. Sawhney, W. Niblack, et al. Query by image and video content: The QBIC system. *IEEE Computer*, pages 23–32, September 1995.

[12] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A Stahel. *Robust Statistics: The Approach based on Influence Functions*. J. Wiley & Sons, NY, 1986.

[13] K. J. Hanna. Direct multi–resolution estimation of ego–motion and structure from motion. In *Proc. IEEE Wkshp. on Visual Motion*, pages 156–162, 1991.

[14] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In *Joint European-US Workshop on Applications of Invariance in Computer Vision*, 1993.

[15] Hideharu Hashihara, Jun ichi Takahashi, and Jung-Kook Hong. Scene retrieval method for motion image databases. Technical report, IBM Tokyo Research Laboratory, 1991.

[16] V. Hasselblad. Estimation of parameters for a mixture of normal distributions. *Technometrics*, 8(3):431–446, August 1966.

[17] Kyoji Hirata and Toshikazu Kato. Rough sketch-based image information retrieval. *NEC R&D*, 34(2):263–273, April 1993.

[18] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representation. In *ICPR*, pages 743–746, Jerusalem, Israel, Oct. 1994.

[19] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proc. Intl. Conf. on Computer Vision*, pages 605–611, 1995.

[20] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV*, pages 282–287, Santa Margherita, Italy, May 1992.

[21] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761, New York, USA, June 1993.

[22] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *ICPR*, pages 685–688, 1994.

[23] R. Kumar, P. Anandan, M. Irani, et al. Representation of scenes from collection of images. In *Proc. IEEE Wkshp. on Representation of Visual Scenes*, 1995. http://www.cis.upenn.edu/∼eero/VisSceneRep95.html.

[24] Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, 1989.

[25] G. Li. Robust regression. In D.C. Hoaglin, F. Mosteller, and J.W. Tukey, editors, *Exploring Data Tables, Trends and Shapes*, chapter 8. John Wiley and Sons, NY, 1985.

[26] W.J. MacLean, A.D. Jepson, and R.C. Frecker. Recovery of egomotion and segmentation of independent object motion using the em algorithm. In *BMVC*, 1994. Submitted paper.

[27] S. Mann and R. W. Picard. Virtual bellows: Constructing high quality stills from video. In *ICIP*, 1994.

[28] G.J. McLachlan and K.E. Basford. *Mixture Models Inference and Applications to Clustering*. Marcel Dekker, Inc., New York and Basel, 1988.

[29] W. Niblack, R. Barber, W. Equitz, et al. The QBIC project: Querying images by content using color, texture, and shape. In *SPIE 1908, Storage and Retrieval for Image and Video Databases*, pages 173–187, Feb. 1993.

[30] J.M. Odobez and P. Bouthemy. Detection of multiple moving objects using multiscale mrf with camera motion compensation. In *ICIP*, pages 257–261, Austin, USA, Nov. 1994.

[31] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models in complex image sequences. In *7th EUSIPCOEuropean Conference on Signal Processing*, pages 411–414, Edinburgh, Scotland, Sep. 1994.

[32] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Proc. Storage and Retrieval for Image and Video Databases II*. SPIE, 1994.

[33] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

[34] P. J. Rousseeuw and A. M. Leroy. *Robust Regression & Outlier Detection*. J. Wiley & Sons, NY, 1987.

[35] H. S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. In *Proc. Intl. Conf. on Pattern Recognition*, pages A403–A408, 1994.

[36] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D&3D dominant motion estimation for mosaicing and video representation. In *Proc. Intl. Conf. on Computer Vision*, pages 583–590, 1995. ftp://eagle.almaden.ibm.com/pub/cs/reports/vision/dominant_motion.ps.Z.

[37] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley, NY, 1989.

[38] A. Shashua and N. Navab. Relative affine structure: Thoery and application to 3D reconstruction from perspective views. In *Proc. Computer Vision & Pattern Recognition Conf.*, pages 483–489, 1994.

[39] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Wkshp. on Applications of Computer Vision*, pages 44–53, 1994.

[40] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *Proc. IEEE Wkshp. on Representation of Visual Scenes*, 1995. http://www.cis.upenn.edu/∼eero/VisSceneRep95.html.

[41] Laura A. Teodosio and Walter Bender. Salient video stills: Content and context preserved. In *ACM Intl. Conf. on Multimedia*, 1993.

[42] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. VideoMAP and VideoSpaceIcon: Tools for anatomizing video content. In *ACM INTERCHI*, pages 131–136, 1993.

[43] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *Proc. Computer Vision & Pattern Recognition Conf.*, pages 361–366, New York, USA, June 1993.

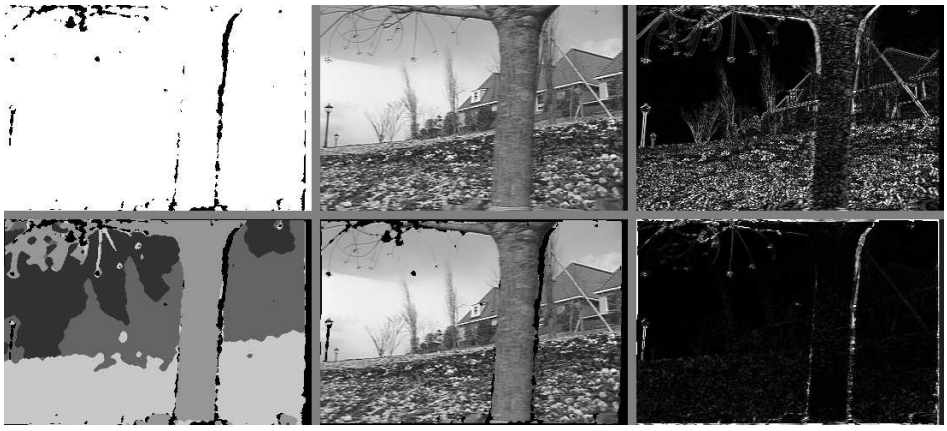

Figure 9: **Middle**: Original frame (top) and synthesized frame (bottom) using warped layers with their motion estimates for the *fg* sequence. **Left**: Outliers (top, in black) and four labeled layers (bottom) after EM-MDL-MAP estimation shown in different shades of grey, outlier layer in black. **Right**: Original difference between frames (top) and difference between the reference frame and the synthesized image (bottom).

Figure 10: Four layers, after EM-MDL estimation, shown individually in white for the *fg* sequence.

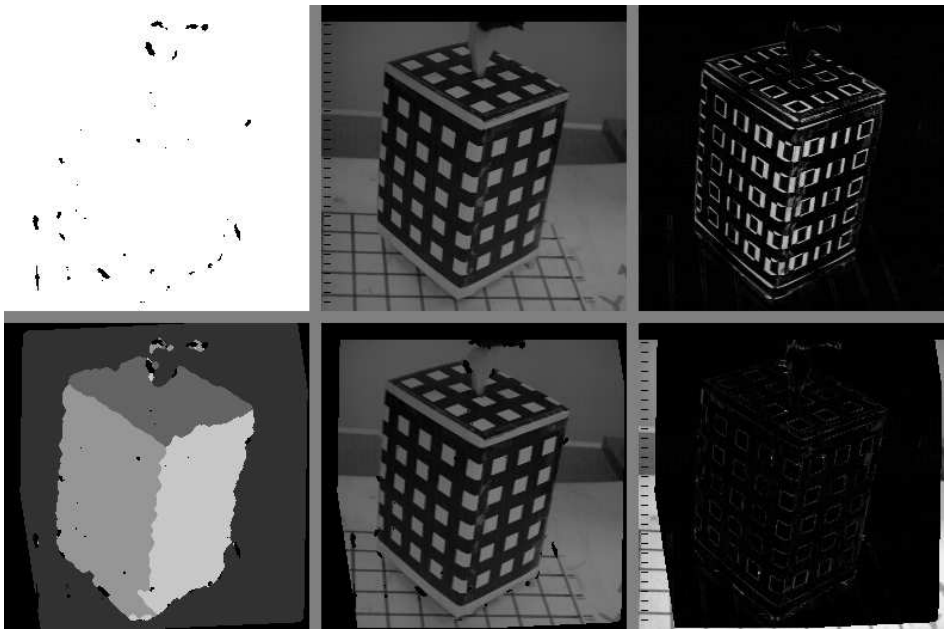

Figure 11: **Middle**: Original frame (top) and synthesized frame (bottom) using warped layers with their motion estimates for the *box* sequence. **Left**: Outliers (top, in black) and four labeled layers (bottom) after EM-MDL-MAP estimation shown in different shades of grey, outlier layer in black. **Right**: Original difference (top) between frames and difference between the reference frame and the synthesized image (bottom).

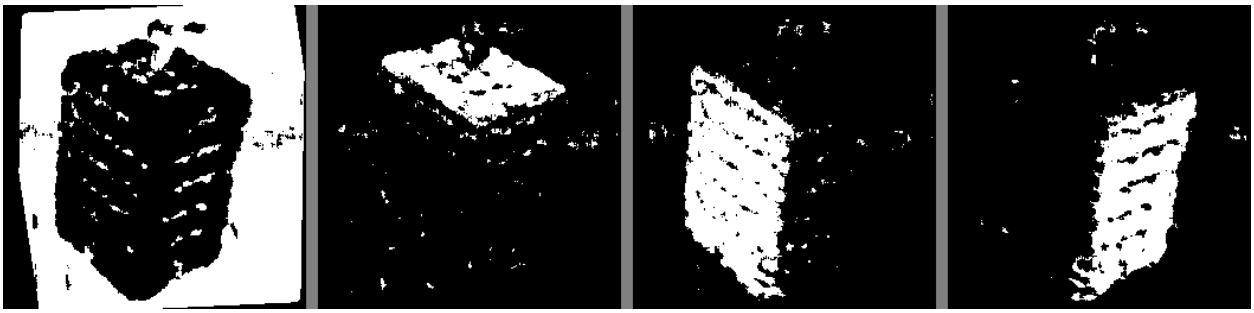

Figure 12: Four layers, after EM-MDL estimation, shown individually in white for the *box* sequence.

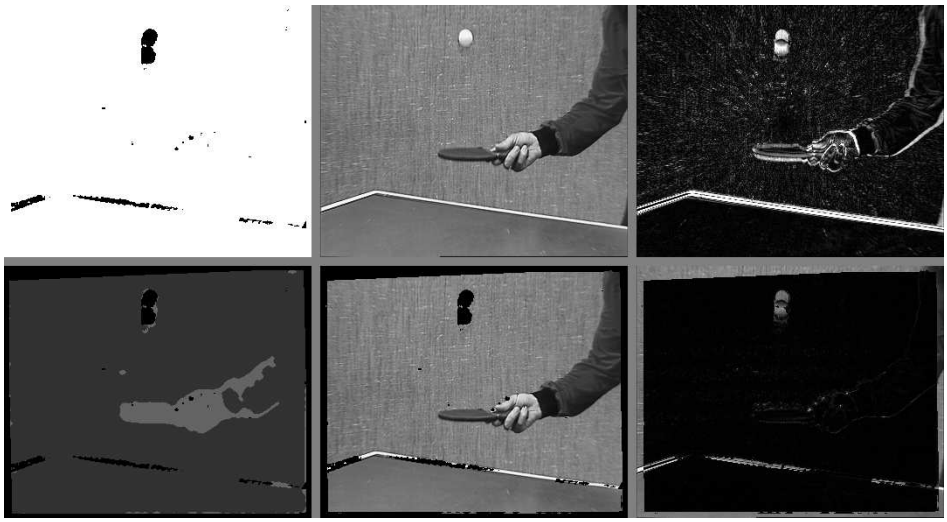

Figure 13: **Middle**: Original frame (top) and synthesized frame (bottom) using warped layers with their motion estimates for the *tt* sequence. **Left**: Outliers (top, in black) and two labeled layers (bottom) after EM-MDL-MAP estimation shown in different shades of grey, outlier layer in black. **Right**: Original difference (top) between frames and difference between the reference frame and the synthesized image (bottom).

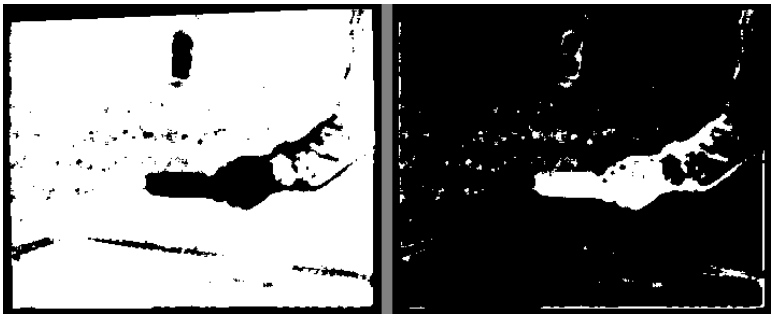

Figure 14: Two layers, after EM-MDL estimation, shown individually in white for the *tt* sequence.