A SHORTER VERSION APPEARED IN THE IEEE INTL. CONF. ON COMPUTER VISION, CAMBRIDGE, MA, USA, JUNE, 1995.

Layered Representation of Motion Video using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding

Serge Ayer[1]

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

Harpreet S. Sawhney

Machine Vision Group
IBM Almaden Research Center, K54
650 Harry Road, San Jose, CA 95120
Email: sawhney@almaden.ibm.com

December 1, 1994

Abstract

Representing and modeling the motion and spatial support of multiple objects and surfaces from motion video sequences is an important intermediate step towards dynamic image understanding. One such representation, called *layered representation*, has recently been proposed. Although a number of algorithms have been developed for computing these representations, there has not been a consolidated effort into developing a precise mathematical formulation of the problem. This paper presents such a formulation based on maximum likelihood estimation of mixture models and the minimum description length (MDL) encoding principle.

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. In order to allow multiple models in the description of image motion, the likelihood function for change in intensity of a pixel is modeled as an additive mixture of Gaussian densities. Robust maximum-likelihood estimation (MLE) of the multiple models, and their layers of support, represented as ownership probabilities, is performed using a modified Expectation-Maximization (EM) algorithm. The adequate number of models is automatically decided using the MDL principle that minimizes the encoding length of the model parameters, and the MLE residuals. Iterative application of the EM algorithm and the MDL principle implemented in a hierarchical, direct estimation framework is shown to generate good layered descriptions in many real motion sequences without the need for any ad-hoc parameters.

[1] 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

Description of objects and surfaces in terms of their spatial support and image or 3D motion is an important intermediate goal in dynamic image understanding. Recently there has been a trend towards building one such representation, called a *layered* description of moving images. A layered description consists of three parts: (i) a set of motion descriptors, (ii) layers of support for each motion descriptor, and (iii) a compact representation of the intensity map for each layer that can be derived from (i) and (ii) and the original video sequence. Each layer corresponds to a set of pixels that move over time according to a model of motion, and are identified over the images in a sequence. For each image in the sequence, there is a layer of ownership weights and the corresponding motion parameters that describe the motion transformation between the layer in this image and the reference image. With the knowledge of the motion parameters and the ownership weights for each layer in each image, a reference intensity map for the layer can be created in the reference frame. This reference map is a single mosaiced frame for a layer that depicts whatever was seen for that layer over the whole sequence. Therefore, the problem to be solved is the computation of the motion descriptors and the layers of ownership weights. In the most general case of motion transparency, the ownership weights are analog, for instance probabilities, and for the common case of opaque motions, binary weights are adequate.

Automatic description of motion video sequences in terms of layers of multiple models has wide-ranging applications. Video compression and coding have already been highlighted in the work of Wang and Adelson [23]. Also, the layered representation can be used as a compact description to be used for matching and recognition, and for synthesizing novel videos from existing ones. Emerging applications in the areas of automatic video annotation and indexing will benefit too from the layered representations. Automatically extracted layers of intensity maps, and the associated weight masks and motion parameters, can be used for annotating a video sequence with its pictorial and motion content. Static image features (like shape, color and texture) computed over the separated layers, and their motion descriptions along with their change over time can then be used for content-based querying and indexing. Video indexing systems in the spirit of static image indexing like in the Photobook [17] and Query-By-Image-Content (QBIC) [6] systems can effectively use layered description as an intermediate representation of video content. A number of algorithms have been developed for computing these representations [1, 5, 9, 10, 24]. However, there has not been a consolidated effort into developing a precise mathematical formulation of the problem. This paper presents such a formulation based on maximum likelihood estimation of mixture models and the minimum description length (MDL) encoding principle.

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. In order to allow multiple models in the description of image motion, we model the likelihood function for change in intensity of a pixel, conditioned on the motion parameters, as an additive mixture of Gaussian densities. This is called a *mixture* model. Robust maximum-likelihood estimation (MLE) of the mixture model, and the layers of support, represented as ownership probabilities, is performed using a modified Expectation-Maximization (EM) algorithm. The adequate number of models is automatically decided using the MDL principle that minimizes the encoding length of the model parameters and the MLE residuals. This is a generalization of the encoding length computed using the log-likelihood of the Maximum A-Posteriori (MAP) estimate [13], with the log-likelihood term for the measurements conditioned on the models being derived using the mixture model formulation. Our algorithm implements the ML estimation using a direct, hierarchical method [2]. Amongst the algorithms that compute layered representations using *simultaneous* (and not *sequential*) multiple motion estimation, ours is the only one that we know of in which support layers at the coarse resolutions are used to guide their estimation at the finer resolutions.

Our solution to the problem of layered representation combines the advantages of direct motion estimation methods, robust estimation, MDL coding, and mixture of models. Current algorithms use only *some* aspects of this formulation. However, we have found, and will demonstrate that a formulation and an algorithm that integrates all these aspects leads to an automatic layered representation for a variety of motion sequences without any ad-hoc parameters.

2 Background

The main problem addressed in this paper is that of computing multiple models of motion and the layers of their spatial support in image sequences. There are two major approaches to this problem. 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. Also, all the algorithms to date use essentially a two-frame formulation for computing the motion parameters and the support layers. The pairwise parameters are used to create intensity maps in a given reference frame. Our algorithm also uses a two-frame formulation.

Wang and Adelson [24] addressed the problem in terms of the computation of affine motion models, and their binary layers of support whose ordering in depth is computed using their support over many frames. The essential idea in their work is that of iteratively clustering motion models computed using pre-computed *optical flow.* The optical flow vectors are assigned to their best motion models, starting with a large initial set of models. Initially k-means clustering is done using a pre-specified k, and then models are merged iteratively, based on a threshold on the difference in motion they specify.

The main drawbacks of the above approach are its use of optical flow as an input representation, clustering in the parameter space, and the use of binary ownership weights throughout the iterative process. In computing the optical flow, algorithms generally make smoothness assumptions that can distort the structure of image motion. Given that they use parametric models of motion, computing motion parameters directly from spatio-temporal intensities would be more robust. They try to correct the errors in optical flow by reverting back to the intensity maps, but this is avoided by the direct approach. 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 layers-of-support estimation within a robust M-estimation and MDL framework [5]. They use a truncated quadratic optimization function, that reduces the weight of residuals beyond a threshold to zero, for estimating the motion parameters. MDL criterion is applied to the encoding of the model parameters and the maximum-likelihood estimates of the residuals.

The important problem of estimating the threshold (or the scale parameter of the influence function) [14] for the truncated quadratic is not addressed in their approach. Moreover, the truncated quadratic is too severe and can reject non-outliers too especially in the initial stages of the iterative process, without a sound method for automatically estimating the threshold. They too compute a binary ownership weight for each measurement based on the model that has the least residual. This is done for each iteration. This puts strong requirements on the initial estimates of the motion parameters and the scale estimates to be very good. They use a 2D similarity transform as their motion descriptor. The output of their algorithm on real image sequences leaves much room for improvement.

Irani and al. [10] addressed the problem of detecting and tracking multiple moving objects over image sequences. However, their approach to the estimation of multiple motions is through successive estimation of dominant motions using model-based least squares (LS) estimation, i.e. it falls in the class of sequential methods. There are two potential problems with this formulation. First, in the absence of competing models, the dominant motion model can irrevocably commit data erroneously to a model. This is even more of a problem in their approach because they do not have any mechanism for automatically estimating the scale parameter of residual errors (akin to the problem with Darrell and Pentland's approach). They apply an absolute threshold to a motion measure defined using normal flow computed at each point, to decide if the point belongs to the model. The motion measure used is in general quite unstable, especially in low gradient regions, and can lead to arbitrary decisions without an adaptive scale/threshold estimate. Second, their use of least squares estimation, and thresholding based on the residuals may lead to arbitrariness in what is called dominant in the data because of the classic problem of masking of outliers in LS estimation [19].

Ayer et al. [1] too adopted a sequential approach to fleshing out multiple models of motion over an image sequence. In contrast with Irani and 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.

Hsu et al. [9] in their work on optical flow computation using layered motion representation have laid out a qualitative framework for such a description. Their algorithms are a collection of algorithms like the ones by Wang and Adelson, and Irani and al. They do not, however, give a firm quantitative formulation of the framework. In this work, we have formally addressed all the issues discussed by them.

Black and Anandan [3] incorporated robust M-estimators in their solution to multiple motion estimation, again through successive separation of dominant motions. They dwell at length on the advantage of using *redescending* estimators and their ability to decrease the influence of outliers as governed by the scale parameter. However, in their algorithms for robust estimation of parametric models, 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. Furthermore, they use an ad-hoc fixed step size in a modified gradient descent like procedure. They do not take advantage of weighted Gauss-Newton/Levenberg-Marquardt type methods for solving M-estimation problems [14, 21]. Also, their dominant motion approach again does not let competing models vie for ownership of the measurements.

MacLean et al. [15] 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 [11] applied the EM algorithm using the mixture model formulation to optical flow computation without estimating the number of models. Also, they used a 2D translational model only.

3 Overview of the formulation

Our approach uses simultaneous estimation of motion models and their layers of support. Motion models compete for the support of pixels and the supports in turn influence the estimation of model parameters. In contrast with existing simultaneous estimation methods, the number of models is decided using both the quality of description of the data and the cost of the models, and the scale estimates that are used to decide the ownership weights are computed automatically within the competitive framework. In contrast with sequential methods, no external threshold is required to decide whether a pixel belongs to a model. No assumptions about the presence of a dominant layer are required.

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, the layers of support, and the residual at each pixel resulting from the difference between the reference intensity map and warped map corresponding to the motion parameters. Solving for all the unknown parameters simultaneously is practically impossible because of the large parameter space. Therefore, 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 motion descriptors used are 2D parametric models: translational, affine and projective. First, the probabilistic model of the reference image as arising from a mixture of densities corresponding to a given number of models, g, is presented in Section 4.1. In particular, normal densities are used. The unknowns in this model are the g sets of motion parameters, the variances of the associated gaussian densities, and the ownership probabilities (layers) of each pixel. Both a continuous ownership probability and its specialization to a binary ownership are presented. It is shown how the necessary conditions for ML estimation of the mixture model leads to equations for solution of the unknowns. This leads to the iterative Expectation-Maximization (EM) algorithm; E step to solve for layers given the motions and the variances, and the M step for solving for the latter given the layers. Next, in Section 4.3, the specific encoding of models and data is presented. A highlight of the encoding is the use of log-likelihood of the *mixture* densities for encoding the data conditioned on the models.

The computation of the motion parameters uses direct methods that model image motion as a change of the coordinate system with brightness constancy. However, instead of using least squares estimation corresponding to the normal densities modeled by the standard EM approach, we use robust M-estimators. This has two advantages: (i) in the early steps of the iterative algorithm when layers and motion models have not yet been computed accurately, the M-estimators allow for the influence of outliers for a model to be reduced, and (ii) influence of data that does not satisfy the brightness constancy assumption is also reduced. Section 5 presents the robust M-estimation formulation, and a solution using Gauss-Newton method.

Estimation of the other mixture parameters, the variances and the layers is the subject of Section 6. The subsequent section is devoted to the details of the complete algorithm. Section 8 presents experimental results and Section 9 wraps up this presentation with a discussion of work in progress and unresolved issues.

We wish to point out that all the current layered representation formulations are essentially two-frame and so is our present work. Using the computed layers and the motion models on a two-frame basis to create a mosaiced intensity map for each layer in a reference view, and also inferring depth ordering in this process are relatively straightforward steps [24]. These are not the focus of this work. However, formulating the problem as inherently a multiple frame estimation problem is the subject of ongoing work. Some of the issues will be discussed in the conclusions section.

4 Mixture model and MDL formulation

4.1 Mixture Models for Maximum-Likelihood estimation

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 of which is warped using its own motion model. The intensity $I(\mathbf{x}_j, t)$ at pixel j 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 $\{\tilde{I}_1, \ldots, \tilde{I}_g\}$. $\tilde{I}_i$ 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,

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

The probability density function (*pdf*) of the reference intensity map, $I(\mathbf{x}, t)$, as predicted from $\tilde{I}$ can be represented in the finite mixture form [16] as,

$$p(I(\mathbf{x}, t) \mid I(\mathbf{x}, t-1), \boldsymbol{\Phi}) = \sum_{i=1}^{g} \pi_i \, p_i(I(\mathbf{x}, t) \mid \tilde{I}_i(\mathbf{x}, \boldsymbol{\theta}_i, t-1), \sigma_i). \tag{2}$$

The vector $\boldsymbol{\Phi}$ represents the vector of all unknown parameters,

$$\boldsymbol{\Phi} = [\boldsymbol{\Pi}^T, \boldsymbol{\Sigma}^T, \boldsymbol{\Theta}^T]^T, \tag{3}$$

with $\boldsymbol{\Pi} = [\pi_1 \ldots \pi_g]^T$, $\boldsymbol{\Sigma} = [\sigma_1 \ldots \sigma_g]^T$, and $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_g]^T$, where $\boldsymbol{\theta}_i$ are the parameters of the ith motion model, σ_i^2 is the variance, and π_i is the proportion of the ith model in the mixture such that,

$$\textstyle\sum_{i=1}^{g} \pi_i = 1, \quad \pi_i > 0, \quad (i = 1, \ldots, g).$$

In order to simplify the above formulation, we want to express the finite mixture form in terms of the prediction error $r_i(\mathbf{x}) = I(\mathbf{x},t) - \tilde{I}_i(\mathbf{x},\boldsymbol{\theta}_i,t)$, for the ith model at pixel location $\mathbf{x}$. Let $\mathbf{r}(\mathbf{x})$ be the g-dimensional vector of residuals for the g models at a point $\mathbf{x}$. The finite mixture form for the residuals can now be written as,

$$p(\mathbf{r}(\mathbf{x}) \mid \boldsymbol{\Phi}) = \sum_{i=1}^{g} \pi_i \, p_i(r_i(\mathbf{x}) \mid \boldsymbol{\theta}_i,\, \sigma_i). \tag{4}$$

(In the following, it is assumed that $I(\mathbf{x},t-1)$ is given and is not included explicitly in the functions, and $I(\mathbf{x},t)$ is written as $I(\mathbf{x})$ unless necessary otherwise.)

We assume that each $p_i(r_i(\mathbf{x}) \mid \boldsymbol{\Phi})$ belongs to the same parametric family (e.g normal distributions). In particular, $p_i(r_i(\mathbf{x}) \mid \boldsymbol{\Phi})$ is assumed to be a normal distribution with zero mean and variance, σ_i^2, $\mathcal{N}(0,\sigma_i^2)$. Under the assumption that the N observations, one at each pixel, are realized values of N independent and identically distributed (i.i.d.) random variables with common distribution function $p(\mathbf{r}(\mathbf{x}) \mid \boldsymbol{\Phi})$, the negative log-likelihood function $L(\boldsymbol{\Phi})$ can be written as,

$$\begin{aligned} L(\mathbf{r}(\mathbf{x}_1),\ldots,\mathbf{r}(\mathbf{x}_N) \mid \boldsymbol{\Phi}) &= -\log(p(\mathbf{r}(\mathbf{x}_1),\ldots,\mathbf{r}(\mathbf{x}_N)) \mid \boldsymbol{\Phi})) \\ &= -\log(\prod_{j=1}^{N} p(\mathbf{r}(\mathbf{x_j}) \mid \boldsymbol{\Phi})) = -\sum_{j=1}^{N} \log(\sum_{i=1}^{g} \pi_i \; p(r_i(\mathbf{x_j}) \mid \boldsymbol{\theta}_i,\, \sigma_i)) \end{aligned} \tag{5}$$

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

$$\begin{aligned} \tau_{ij} &= p(\mathbf{x}_j \in \boldsymbol{\phi}_i \mid r(\mathbf{x_j}); \boldsymbol{\Phi}) \\ &= \frac{p(\mathbf{x}_j \in \boldsymbol{\phi}_i \mid \boldsymbol{\Phi}) \; p(r(\mathbf{x}_j) \mid \mathbf{x}_j \in \boldsymbol{\phi}_i; \boldsymbol{\Phi})}{p(r(\mathbf{x_j}); \boldsymbol{\Phi})} \\ &= \frac{\pi_i \; p(r_i(\mathbf{x_j}) \mid \boldsymbol{\theta}_i, \sigma_i)}{\sum_{i=1}^{g} \pi_i \; p(r_i(\mathbf{x_j}) \mid \boldsymbol{\theta}_i,\, \sigma_i)}. \end{aligned} \tag{6}$$

It can been shown ([16]) that the maximum likelihood estimates of $\boldsymbol{\Phi}$, $\hat{\boldsymbol{\Phi}}$, in terms of the ownership weights, satisfy

$$\hat{\pi}_i = \sum_{j=1}^{N} \frac{\hat{\tau}_{ij}}{N}, \qquad i = 1,\ldots,g \tag{7}$$

$$\sum_{i=1}^{g}\sum_{j=1}^{N}\hat{\tau}_{ij}\frac{\partial \log(p(r_i(\mathbf{x}_j) \mid \hat{\boldsymbol{\theta}}_i, \hat{\sigma}_i))}{\partial \hat{\sigma}_i} = 0 \tag{8}$$

$$\sum_{i=1}^{g}\sum_{j=1}^{N}\hat{\tau}_{ij}\frac{\partial \log(p(r_i(\mathbf{x}_j) \mid \hat{\boldsymbol{\theta}}_i, \hat{\sigma}_i))}{\partial \hat{\boldsymbol{\theta}}_i} = 0. \tag{9}$$

4.2 Specialization to binary ownerships

In accordance with the formulation in [16, pg. 14], the negative log likelihood function like the one in equation (5) is now presented for the case when each measurement is mutually exclusively associated with only one 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,

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

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*, $p(\mathbf{r}(\mathbf{x_j}); \boldsymbol{\theta} \mid \mathbf{z_j})$, is given by,

$$p(\mathbf{r}(\mathbf{x_j}); \theta \mid \mathbf{z_j}) = \prod_{i=1}^{g} [\pi_i \, p(r_i(\mathbf{x_j}) \mid \boldsymbol{\theta}_i, \sigma_i)]^{z_{ij}}.$$

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,

$$L_C(\mathbf{r}(\mathbf{x}_1), \ldots, \mathbf{r}(\mathbf{x}_N); \boldsymbol{\Phi} \mid \mathbf{z_1}, \ldots, \mathbf{z_N}) = -\sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij} \left\{\log \pi_i + \log p(r(\mathbf{x_j}) \mid \boldsymbol{\theta}_i, \sigma_i)\right\} \tag{10}$$

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.

Equations (7)–(9) and equation (6) suggest an iterative solution for the maximum likelihood estimates of $\boldsymbol{\Pi}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\Theta}$, and for the posterior ownership probabilities, τ's. One solution method is called the Expectation-Maximization (EM) algorithm. The M step is for the ML estimates of the model parameters, and the E step for the ownership probabilities. Under the assumption of normal densities for p_i's, the solutions for the π's, σ's and the τ's are presented in section 6. For the estimates of the motion parameters, $\boldsymbol{\theta}_i$'s, a modified M step is used to allow the use of robust M-estimator functions. This is the subject of section 5. Presently, we turn to the problem of determining modeling complexity using MDL.

4.3 MDL formulation for determining model complexity

The problem with the maximum likelihood formulation of equations (7)–(9) 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. A second important reason is that the MDL principle leads to an objective function with no arbitrary thresholds. 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 encoding length.

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

$$\mathcal{L}(\mathbf{r}(\mathbf{x}_1), \ldots, \mathbf{r}(\mathbf{x}_N), \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) + \mathcal{L}_D(\mathbf{r}(\mathbf{x}_1), \ldots, \mathbf{r}(\mathbf{x}_N) \mid \mathbf{\Phi}), \tag{11}$$

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 from (3). Thus,

$$\mathcal{L}_M(\mathbf{\Phi}) = \mathcal{L}_{M1}(\mathbf{\Pi}) + \mathcal{L}_{M2}(\mathbf{\Sigma}) + \mathcal{L}_{M3}(\mathbf{\Theta}). \tag{12}$$

For computing the coding cost of these real-valued parameters, the expression derived by Rissanen [18] 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 $\mathcal{L}_D(\mathbf{r}(\mathbf{x}_1), \ldots, \mathbf{r}(\mathbf{x}_N) \mid \mathbf{\Phi})$, i.e. to encode the residuals. Since we know the probability, $P(\mathbf{r}(\mathbf{x}) \mid \mathbf{\Phi})$, from the mixture model, the optimal number of bits required to encode this is just the negative logarithm of the probability [18]. Therefore, this term is directly derived from the negative log-likelihood of the data given the model, presented in equations (5) and (10), by replacing the *pdf* $p(r(\mathbf{x_j}) \mid \theta_\mathbf{i}, \sigma_i)$ by the corresponding probability $P(r(\mathbf{x_j}) \mid \theta_\mathbf{i}, \sigma_i)$. Under the assumption of normal distribution of the residuals, and if the residuals are quantized to the nearest ϵ, their real precision, then [13]

$$P(r(\mathbf{x_j}) \mid \theta_\mathbf{i}, \sigma_i) \approx \frac{\epsilon}{\sqrt{2\pi}\sigma_i}\exp(\frac{-r_i^2(\mathbf{x_j})}{2\sigma_i^2}), \quad \text{when} \;\; \epsilon < \sigma_i.$$

Therefore, by substituting this in equations (5) and (10) for the non-binary and binary likelihoods, the total encoding length is given by,

$$\mathcal{L}_{NB}(r(\mathbf{x}_1), \ldots, r(\mathbf{x}_N), \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) - \sum_{j=1}^{N}\log(\sum_{i=1}^{g}\pi_i\frac{\epsilon}{\sqrt{2\pi}\sigma_i}\exp(\frac{-r_i^2(\mathbf{x_j})}{2\sigma_i^2}))$$

$$\mathcal{L}_B(r(\mathbf{x}_1), \ldots, r(\mathbf{x}_N), \mathbf{\Phi}, \mathbf{Z}) = \mathcal{L}_M(\mathbf{\Phi}, \mathbf{Z}) - \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\{\log\pi_i + \log\frac{\epsilon}{\sqrt{2\pi}\sigma_i}\exp(\frac{-r_i^2(\mathbf{x_j})}{2\sigma_i^2})\}$$

These can be simplified, by eliminating the constant terms, to,

$$\mathcal{L}_{NB}(r(\mathbf{x}_1), \ldots, r(\mathbf{x}_N), \mathbf{\Phi}) = \mathcal{L}_M(\mathbf{\Phi}) - \sum_{j=1}^{N}\log(\sum_{i=1}^{g}\pi_i\frac{1}{\sigma_i}\exp(\frac{-r_i^2(\mathbf{x_j})}{2\sigma_i^2})) \tag{13}$$

$$\mathcal{L}_B(r(\mathbf{x}_1), \ldots, r(\mathbf{x}_N), \mathbf{\Phi}, \mathbf{Z}) = \mathcal{L}_M(\mathbf{\Phi}) + \sum_{i=1}^{g}\sum_{j=1}^{N} z_{ij}\{-\log\pi_i + \log\sigma_i + \frac{r_i^2(\mathbf{x_j})}{2\sigma_i^2}\} \tag{14}$$

Both equations (13) and (14) 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 obviously will be prohibitively expensive given the enormous parameter space of ownership weights for the measurements. In order to circumvent this practically impossible task, we divide the problem into alternating steps of ML estimation of the 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. In addition, for the binary case, a local Markov Random Field (MRF) smoothness model is used for the indicator variables, z's, to incrementally update the ownership layers. The details of this algorithmic procedure are presented in section 7.

5 Robust Model-Based Motion Estimation

In the iterative solution of the mixture model parameters introduced in section 4, one of the M-steps is to compute the motion parameters $\mathbf{\Theta}$, given the current estimates of the variances, $\mathbf{\Sigma}$, mixture proportions $\mathbf{\Pi}$, ownership weights τ's ($\mathbf{z}$'s), and motion parameters. This corresponds to a solution of the necessary condition for the ML estimate of $\mathbf{\Theta}$ of equation (9). The probability distributions of the residuals given the parameters are modeled as a mixture of Gaussians. However, in order to allow for outlying data, instead of least squares (LS) estimation of the parameters, we use robust M-estimators. 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. 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.

For simplicity, in the subsequent part of this section the estimation problem only for a particular model i is considered, so the indices corresponding to the motion model will be dropped from the notation. In this framework, the motion model estimation is posed as the minimization of a robust objective function ρ. Within the context of mixture models, this implies that the following function needs to be minimized,

$$h(\boldsymbol{\theta}) = \sum_{j=1}^{N} \tau_j \, \rho(r(\mathbf{x_j}; \boldsymbol{\theta}); \sigma)) \tag{15}$$

where the τ_j (z_j) corresponds to τ_{ij} (z_{ij}), the ownership weight of the jth pixel for the ith model[2]. If the function ρ is taken to be the negative of the log-likelihood function of p in equation (9) (and assuming the measurements errors are independent), the estimates of $\boldsymbol{\theta}$ obtained by minimizing $h(\boldsymbol{\theta})$ correspond to the maximum likelihood estimates given by solving (9). However, here the assumption is that p in equation (9) is normal, but the M-estimation is done through a robust estimator rather than the LS estimator corresponding to the normal density. (For a detailed account of robust estimators, see [8] and [14].)

We have experimented with two ρ functions, the *Lorentzian* and the *Geman & McClure* function [3]:

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

The ρ_{LO} and ρ_{GM} functions give non-zero ascending and descending weights that are controlled by the scale parameters. They differ in how much of an influence any residual has on the estimate. In contrast, LS estimation gives a monotonically increasing linear weights for the residuals, thus, outliers have a strong influence on the estimated parameters. Other M-estimators like Andrew's sine wave and Tukey's Biweight [8] function have also been proposed. However, for both, the weight of residuals beyond a certain threshold reduces to zero. This is not appropriate for our problem, where good initial estimates may not be available, because then potentially sound measurements may be given no weight. This is especially critical in the initial stages of the estimation process when both the motion parameters and the scale estimate have not settled to stable enough values.

The M-estimator for the parameters $\boldsymbol{\theta}$, based on the ρ function in the minimization of $h(\boldsymbol{\theta})$, is the $\boldsymbol{\theta}$ that is a solution of the K equations [14],

$$\sum_i \psi(r(j)) \frac{\partial r(j)}{\partial \boldsymbol{\theta}_k} = 0, \qquad \psi(r) = \frac{\rho(r)}{r}, \qquad k \ = \ 1 \ldots K,$$

where ψ, the derivative of ρ is called the influence function, $r(j)$ is a short form for $r(\mathbf{x_j})$, and

[2] For ease of presentation, we now use only the τ's in the formulation. The specialization to z's for the binary case is obvious.

K is the number of unknown parameters, the dimension of $\boldsymbol{\theta}$. However, instead of solving this non-linear system of equations, an alternative is to apply the Gauss-Newton method to the original minimization problem. With the introduction of a particular approximation, detailed in the next section, this leads to an iterated reweighted least squares method (IRLS). This estimation is an alternative form of M-estimation and is also called W-estimation [14].

5.1 Gauss-Newton formulation for M-estimation of parameters

Gauss-Newton (GN) (and Levenberg-Marquardt LM) method for parameter estimation is an approximation to the general Newton's method, in which the second order term in the Hessian of the error function is ignored. In the GN method, given a solution, $\boldsymbol{\theta}^{(m)}$ at the mth step, the descent direction, $\delta\boldsymbol{\theta}^{(m)}$, is given by

$$\delta\boldsymbol{\theta}^{(m)} = -\mathbf{H}^{-1}(\boldsymbol{\theta}^{(m)})\mathbf{g}(\boldsymbol{\theta}^{(m)}), \tag{16}$$

and

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} + \alpha\delta\boldsymbol{\theta}^{(m)} \tag{17}$$

for some positive α. $\mathbf{H}(\boldsymbol{\theta}^{(m)})$ is the approximation to the Hessian of the error function in (15), involving only the first derivatives of the residuals, and $\mathbf{g}(\boldsymbol{\theta}^{(m)})$ is its gradient, both defined at the current $\boldsymbol{\theta}^{(m)}$. Writing the $\mathbf{g}$ and $\mathbf{H}$ in terms of ρ and $r(j)$, we get,

$$\mathbf{g}_k = \sum_j \tau_j \frac{\partial \rho}{\partial r(j)} \frac{\partial r(j)}{\partial \boldsymbol{\theta}_k} \qquad \mathbf{H}_{kl} = \sum_j \tau_j \frac{\partial^2 \rho}{\partial r^2(j)} \frac{\partial r(j)}{\partial \boldsymbol{\theta}_k} \frac{\partial r(j)}{\partial \boldsymbol{\theta}_l}, \tag{18}$$

as the kth and the klth elements of $\mathbf{g}$ and $\mathbf{H}$, respectively. Thus, $\delta\boldsymbol{\theta}$ can be written in terms of these components as the solution of K linear equations

$$\sum_l \mathbf{H}_{kl}\delta\boldsymbol{\theta}_l = -\mathbf{g}_k, \quad k,l \ = \ 1\ldots K,$$

where K is the number of unknown parameters, the dimension of $\boldsymbol{\theta}$.

However, with the non-quadratic ρ's, $\frac{\partial^2\rho}{\partial r^2(j)}$ can be negative, therefore the solution of (16) may not be a descent direction. For instance, $\frac{\partial^2\rho}{\partial r^2}$ for the two robust M-estimators, ρ_{LO} and ρ_{GM}, respectively are

$$\frac{4\sigma^2 - 2r^2}{(2\sigma^2 + x^2)^2} \qquad \& \qquad \frac{2\sigma^2(\sigma^2 - 3x^2)}{(\sigma^2 + x^2)^3}.$$

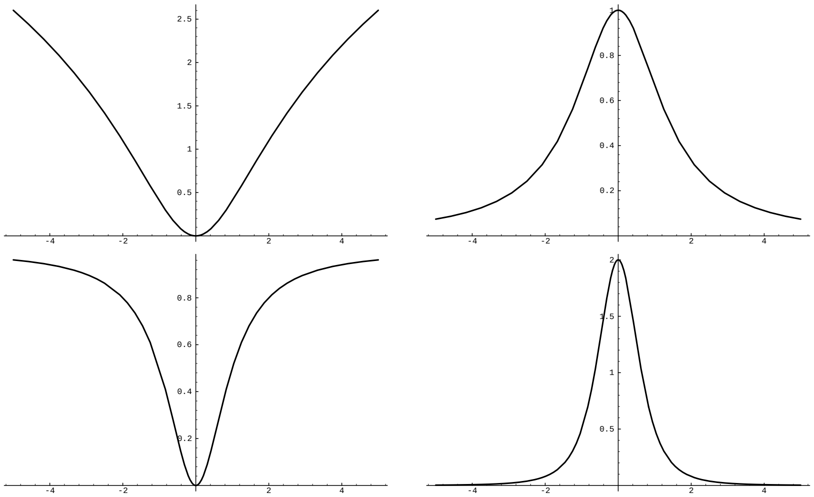

Figure 1: **Left column:** Lorentzian and Geman & McClure functions. **Right column:** The corresponding weights.

If we approximate $\ddot{\rho}(r)$ with its secant approximation [21, pg. 652], $\frac{\dot{\rho}(r)}{r}$, which is positive everywhere, then the GN equations become,

$$\sum_l \sum_j \tau_j \frac{\dot{\rho}(r(j))}{r(j)} \frac{\partial r(j)}{\partial \boldsymbol{\theta}_k} \frac{\partial r(j)}{\partial \boldsymbol{\theta}_l} \delta \boldsymbol{\theta}_l = -\tau_j \frac{\dot{\rho}(r(j))}{r(j)} r(j) \frac{\partial r(j)}{\partial \boldsymbol{\theta}_k}, \quad k, l \;=\; 1 \ldots M, \;\; i \;=\; 1 \ldots N. \tag{19}$$

It is apparent that these equations for the robust objective functions are simply weighted LS normal equations with the weight for each measurement j being $\frac{\dot{\rho}(r(j))}{r(j)}$. It is to be emphasized that in the applications of M-estimators to many vision problems, this connection for the GN formulation has been left unspecified. For the sake of clarity, we have made it explicit in this exposition.

In order to get a feel for the relative weights associated with the various estimators, the plots of the ρ functions for $\sigma = 1.0$, and those of the weights, $\frac{\dot{\rho}(r)}{r}$, are shown in figure 1.

$$\frac{\dot{\rho}_{LO}(r)}{r} = \frac{2}{2\sigma^2 + x^2} \qquad \frac{\dot{\rho}_{GM}(r)}{r} = \frac{2\sigma^2}{(\sigma^2 + x^2)^2} \tag{20}$$

It is apparent from the plots that whereas LS weights residuals of all magnitudes uniformly, ρ_{LO} and ρ_{GM} decrease the influence of large residuals rapidly, ρ_{GM} more so than ρ_{LO}. The parameter σ that controls the location of the inflection point in the curves, governs the point beyond which there is a faster decrease in the influence.

The application of the above formulation to the case of a 2D affine transformation is illustrated now. The flow field $\mathbf{u}(\mathbf{x};\, \boldsymbol{\theta})$ can be written as,

$$\mathbf{u}(\mathbf{x}(x,y);\boldsymbol{\theta}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \boldsymbol{\theta} \;=\; \mathbf{M}\boldsymbol{\theta}.$$

Given an estimate of the parameters $\boldsymbol{\theta}^m$, image $I(t-1)$ is warped so that $I^w(\mathbf{x};\boldsymbol{\theta}^m) = I(\mathbf{x} - \mathbf{u}(\mathbf{x};\boldsymbol{\theta}^m))$. At this step, the residual r at $\mathbf{x}$ is defined as

$$r = I(\mathbf{x} + \delta\mathbf{u}(\mathbf{x};\boldsymbol{\theta}), t) - I^w(\mathbf{x};\boldsymbol{\theta}^m). \tag{21}$$

At the mth iteration, $\delta\mathbf{u}(\mathbf{p};\boldsymbol{\theta}) = \mathbf{M}\delta\boldsymbol{\theta}$. Therefore, in each iteration the derivative of each residual of equation (21) w.r.t. the unknown $\boldsymbol{\theta}$ is,

$$\frac{\partial r}{\partial \boldsymbol{\theta}} = \frac{\partial \delta\mathbf{u}}{\partial \boldsymbol{\theta}} \frac{\partial r}{\partial \delta\mathbf{u}} = \mathbf{M}^T \nabla I(t).$$

This when combined with equations (20) and (19) leads to a new GN direction $\delta\boldsymbol{\theta}$ in each iteration. A line search along this direction is performed to get the local minimum solution for the current iteration (that is α of equation (17)).

6 Estimation of the mixture parameters

With the component densities of the mixture assumed to be normal, the likelihood equations (7) and (8) can be used for an iterative computation of the solution by the EM algorithm [16, pg. 38]. Given estimates of the parameters $\boldsymbol{\Theta}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\Pi}$, in the E step, the posterior probability that the jth pixel belongs to model i, i.e. τ_{ij}, is computed as,

$$\tau_i(r(j)\,|\,\boldsymbol{\Phi}) = pr(j\text{th pixel } \in \tilde{\mathrm{I}}_{\mathrm{i}} \mid \mathrm{r(j)}, \phi_{\mathrm{i}}) = \frac{\pi_{\mathrm{i}}\ \mathrm{p(r(j)} \mid \phi_{\mathrm{i}})}{\sum_{\mathrm{t=1}}^{\mathrm{g}} \pi_{\mathrm{t}}\ \mathrm{p(r(j)} \mid \phi_{\mathrm{t}})} = \frac{\frac{\pi_{\mathrm{i}}}{\sigma_{\mathrm{i}}} \exp(-\frac{\mathrm{r}_{\mathrm{j}}^2}{2\sigma_{\mathrm{i}}^2})}{\sum_{\mathrm{t=1}}^{\mathrm{g}} \frac{\pi_{\mathrm{t}}}{\sigma_{\mathrm{t}}} \exp(-\frac{\mathrm{r}_{\mathrm{j}}^2}{2\sigma_{\mathrm{t}}^2})} \tag{22}$$

The M step consists of solving the likelihood equations (7), (8) and (9) with each $\hat{\tau}_{ij}$ replaced by its value computed in the E-step. Equation (7) already presents the solution for $\hat{\pi}_i$. This step for the ML estimate of the motion parameters, $\boldsymbol{\Theta}$, has already been detailed in section 5. For the case of Gaussian distributions, the solution to the σ's is given by

$$\hat{\sigma}_i = \sum_{j=1}^{N} \frac{\hat{\tau}_{ij} r^2(j)}{N \hat{\pi}_i} \tag{23}$$

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 $(\mathbf{\Pi}, \mathbf{\Sigma})^{(m+1)}$.

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 binary case, we have found that an estimate derived from the median value of the absolute residuals works well. Given contaminated random samples from a Gaussian distribution with a given σ, a robust estimate of σ is related to the samples through

$$\hat{\sigma}_i = 1.4826 \operatorname*{median}_j |r_j|.$$

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 [19, pg. 202] $0.6745 = 1/1.4826$. 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.

An extension of robust M-estimation to the mixture components in the non-binary case can be used by taking into account the weights associated with the GN estimation using the robust ρ functions as was done for the motion parameters in section 5.1 [16]. The robust estimates of $\mathbf{\Pi}$ remain the same, and those for $\mathbf{\Sigma}$ are

$$\hat{\sigma}_i = \sum_{j=1}^{N} \frac{\hat{\tau}_{ij} \frac{\dot{\rho}(r(j);\sigma_i)}{r(j)} r^2(j)}{N_i} \quad \text{where} \quad N_i = \sum_{j=1}^{N} \hat{\tau}_{ij} \frac{\dot{\rho}(r(j);\sigma_i)}{r(j)} \tag{24}$$

Again, the weights are chosen to be $\frac{\dot{\rho}(r;\sigma)}{r}$ for the Lorentzian or Geman & McClure functions.

7 The Algorithm

In Figure 2, we show 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. Recall that the algorithm is implemented in a multi-resolution framework, where the multi-resolution representation can be either a Gaussian or a Laplacian pyramid.

The initialization step consists of the generation of initial estimates for the motion parameters and the σ's. In order to obtain these estimates, rectangular tiled binary layers that cover the entire

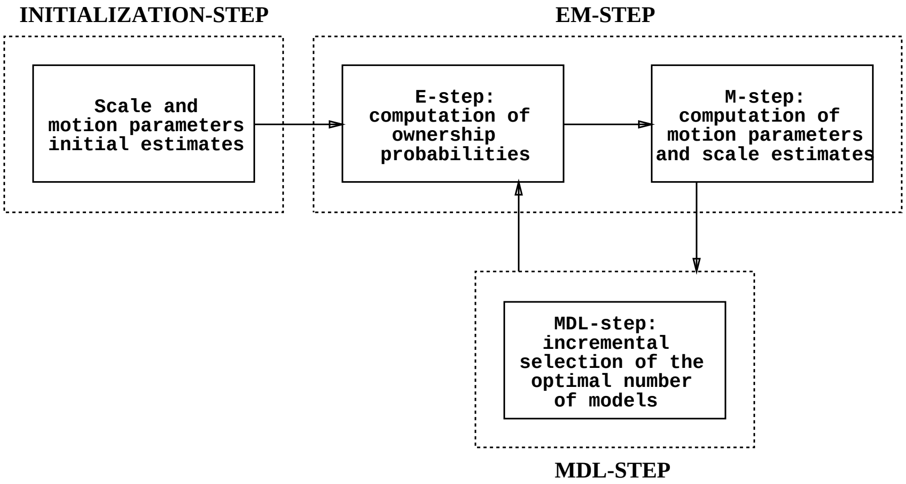
INITIALIZATION-STEP
EM-STEP
Scale and motion parameters initial estimates
E-step: computation of ownership probabilities
M-step: computation of motion parameters and scale estimates
MDL-step: incremental selection of the optimal number of models
MDL-STEP

Figure 2: Flow chart of the algorithm

image are defined. The number of initial layers 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 5. 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, the σ's and the number of models, the ownership weights (τ's/$\mathbf{z}$'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, the model proportions π's, and of the new motion parameters using the new support layers.

The last 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. This computation is the MDL step. 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.

We should mention that the binary method has been tested extensively whereas the non-binary version algorithm is currently under testing.

We note the following important points:

- At the coarsest resolution level, the support layers are computed afresh after each M and MDL step. The rationale is that at the coarse level, in the initial steps, the different parameter estimates may not be good enough. So, in this case, the layer estimates will be changing rapidly, and a scheme which would just update the support layers would probably lead to a non-optimal solution (with a higher number of layers). For the non-binary case, this strategy has been adopted at all levels, and preliminary experiments show that, in order to get the correct number of models, some smoothness constraints need to be introduced. However, we will show that even in the case where the number of model is too high, the motion representation is still very good.

- For the binary case, at the other resolution levels, instead of computing the support layers afresh after the M and MDL steps, a local optimization step is introduced for updating the support layers. The initial estimates of the support layers at a given level are the ones projected down from the next coarser level. The local optimization step implements an MRF based smoothness criterion in the estimation of support layers. A first order 2-clique neighborhood is defined [7]. In a scan-line order, we perform a local minimization at each point following equation (14). In other words, this step tries to find the locally optimal labeling by taking into account both the residual error and the labeling cost. The label that results in the largest positive decrease for the encoding length is committed. This local optimization step is in addition to the global MDL layer removal step described above. The rationale behind the local step is twofold: (i) relatively low gradient regions at the finer levels may have unstable labeling, whereas the corresponding low resolution regions may still be unambiguous, (ii) after convergence at the coarse levels, the layer supports should not change radically but only incrementally. No other simultaneous algorithm uses the projection of layer estimates from coarse levels to guide the solution at the finer level. This is important for stable layer estimation.

- The residual error images need to be filtered for the algorithm to converge to a good number of layers. Median filtering over a 3×3 region around each pixel is performed in our experiments.

We also perform outlier detection by removing all pixels which are atypical of each component of the mixture. This detection is performed by thresholding the residuals using a 2.5 factor of each computed σ's. This allows us to detect pixels which violate the brightness constancy constraint employed in the direct method for motion estimation (i.e. motion of highlights, occlusion/deocclusion intensity patterns).

8 Results

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 multiple moving objects and is also general enough to deal with scenes with moving or static cameras. In showing the results, it is to be emphasized that the results are only an internal representation of the motion and spatial supports, and thus should be considered as an intermediate step towards dynamic image representation. For the different sequences, two frames from the original sequence are shown. Also shown are the labeled layers and the outliers along with the residual errors between the reference image and the second image warped using each of the motion models for each layer.

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 may end up finding some average motion parameters for some layers. However, our simultaneous competitive method leads to reasonably meaningful layer description because measurements are allowed to choose the best model amongst a few.

In showing the results of the algorithm, it is to be emphasized that the results shown in print are nowhere near as dramatic as when shown as moving images. There are two important issues in the quality of the results: (i) the quality of the computed motion measured in terms of how well it is able to compensate or fixate the corresponding layer through warping, and (ii) the quality of the internal representation of either a binary or weighted masks for the layers. Unfortunately, the first cannot be depicted easily on paper, but when the compensated images are shown as a sequence on a screen, the effect is dramatic. On paper, this is shown as residual images.

The first results of layered representation are shown on a synthetic image sequence of moving random dot patterns. This sequence is composed of four translating patterns, whose size is fixed over time. Thus, at the boundaries between two square regions, there may be covered/recovered

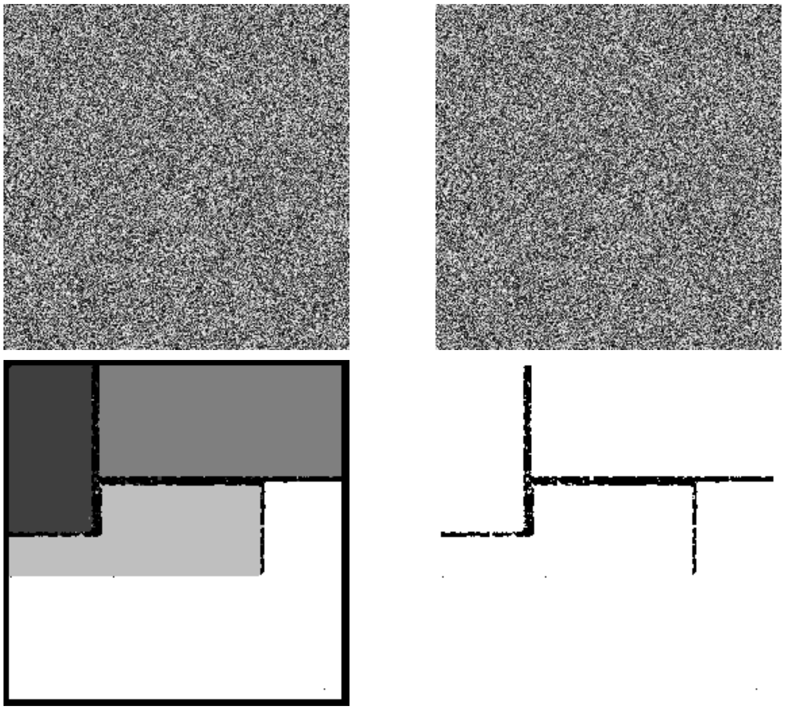

Figure 3: Original images, layers of motion and outliers for the random dot pattern sequence

pixels, depending on the direction of translation of the two regions. Figure 3a) shows two frames of the sequence, the labeled layers, and the outliers. This figure shows that our algorithm captures the four different regions as well as the covered/recovered regions, whose pixels are atypical of each component of the mixture and modeled as outliers.

The next results of layered representation are shown on a sequence showing a box rotating around its vertical axis. In this scene, the camera is static, and hence so is the background in the images. Figure 4a) shows two frames of the *box* sequence, the labeled layers, and the outliers. Figure 4b) shows the residual error associated with each of the layers shown in figure 4a). The results show that the background layer with zero motion and the three faces of the box have been correctly separated.

The next image sequence captures a different 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. Figure 5a) shows the results for the binary layers and figure 5b), shows the associated residual error images. Note that the occlusion region at the right edge of the tree is well captured as outliers.

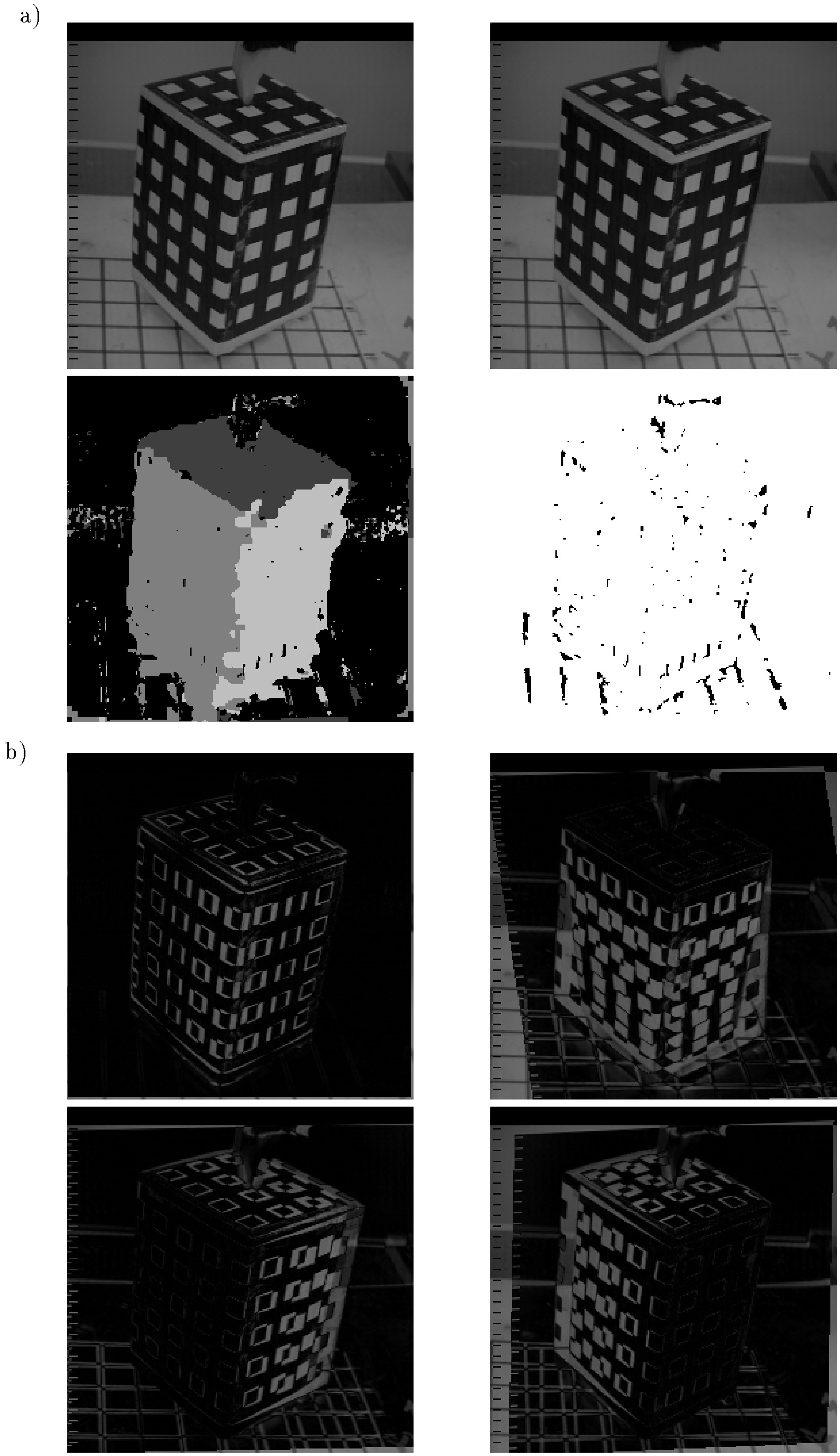

Figure 4: a) Original images, layers of motion and outliers b) Residual errors associated with each layer for the box sequence

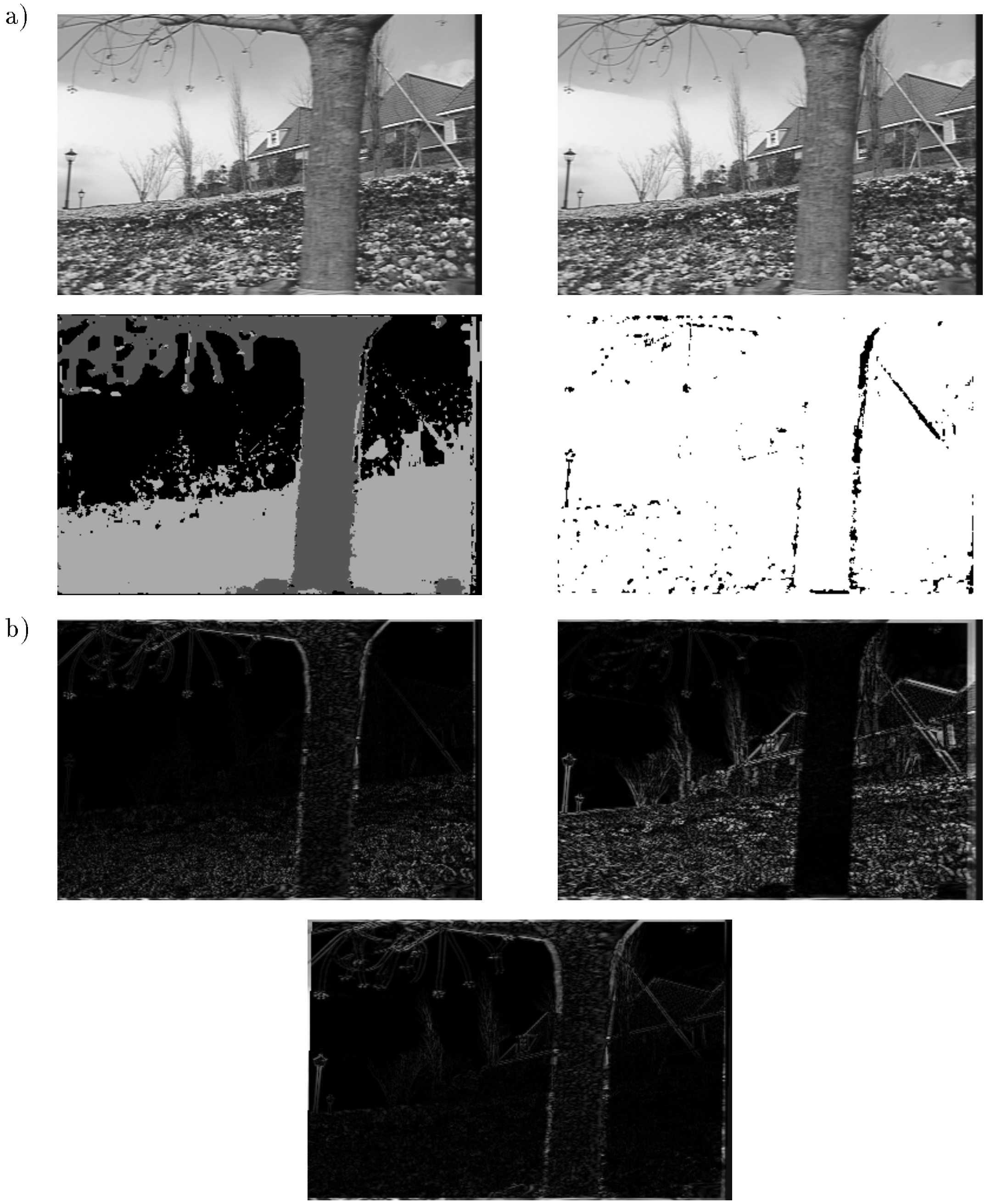

Figure 5: a) Original images, layers of motion and outliers b) Residual errors associated with each layer for the flower garden sequence

To illustrate the situation where both the camera and the objects are moving, results on a sequence of table tennis play are shown. In this sequence, the camera zooms in and the hand of the player moves up tossing the ball. The zoom induces a motion of almost $8 - 10$ pixels at the periphery. 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 figure 6. Due to the fine textured background, interpolation for warping and some systematic jitter/noise around the edges of the table, some pixels in the background are not completely differenced out in the residual image.

Another example where both the camera and the objects are moving is the mobile and calendar sequence. In this sequence, the camera is panning while four objects are moving independently (the train, the rotating toy, the dice, and the calendar). Figures 7 show that the method delivers three layers representing the different regions. The rotating toy and the dice are not labeled as separate layers because their support seems to be too small.

9 Conclusions

The focus of this work has been an automatic computation of a compact, layered description of motion video sequences. Each layer corresponds to a specific parametric model of image motion and has an associated ownership weights for the pixels in the layer. For compact descriptions, the need is to find a small number of models that adequately describe image motion in the sequence with minimal residuals between the motion predicted intensity maps and the observed ones.

There are tradeoffs between the two major approaches to computing such a layered description. The simplicity of formulation and the associated algorithm for a sequential 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. Also, sequential methods rely on the presence of a dominant, and some secondary layers in the data. This assumption maybe a reasonable one if 3D models that can model the image motion of the whole background are used. However, since methods have largely used global 2D parametric models, the absence of a dominant layer corresponding to a parametric model may easily mislead the sequential algorithms.

Simultaneous estimation of the motion parameters and their layers of support adds to the

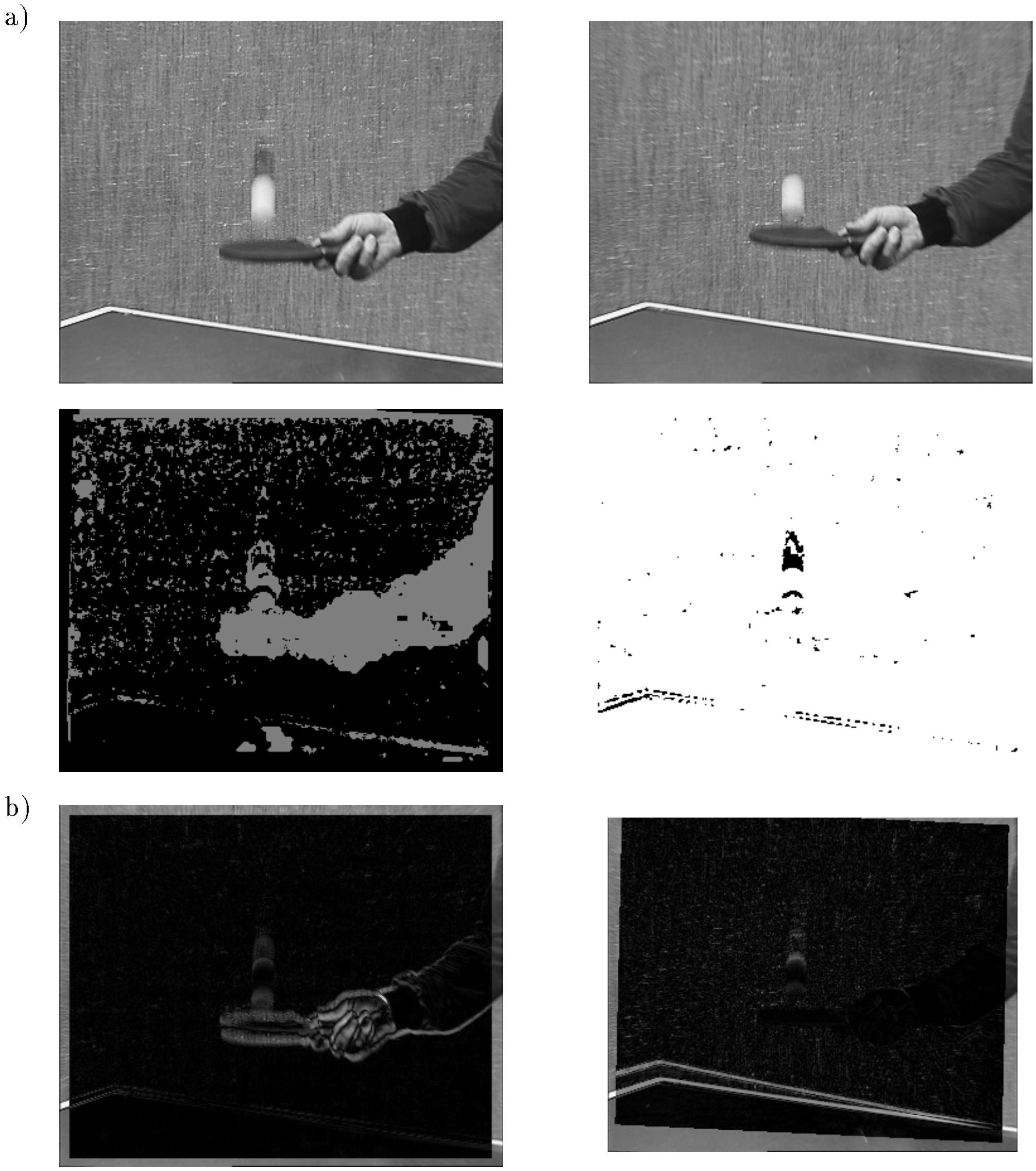

Figure 6: a) Original image, layers of motion and outliers b) Residual errors associated with each layer for the table tennis sequence

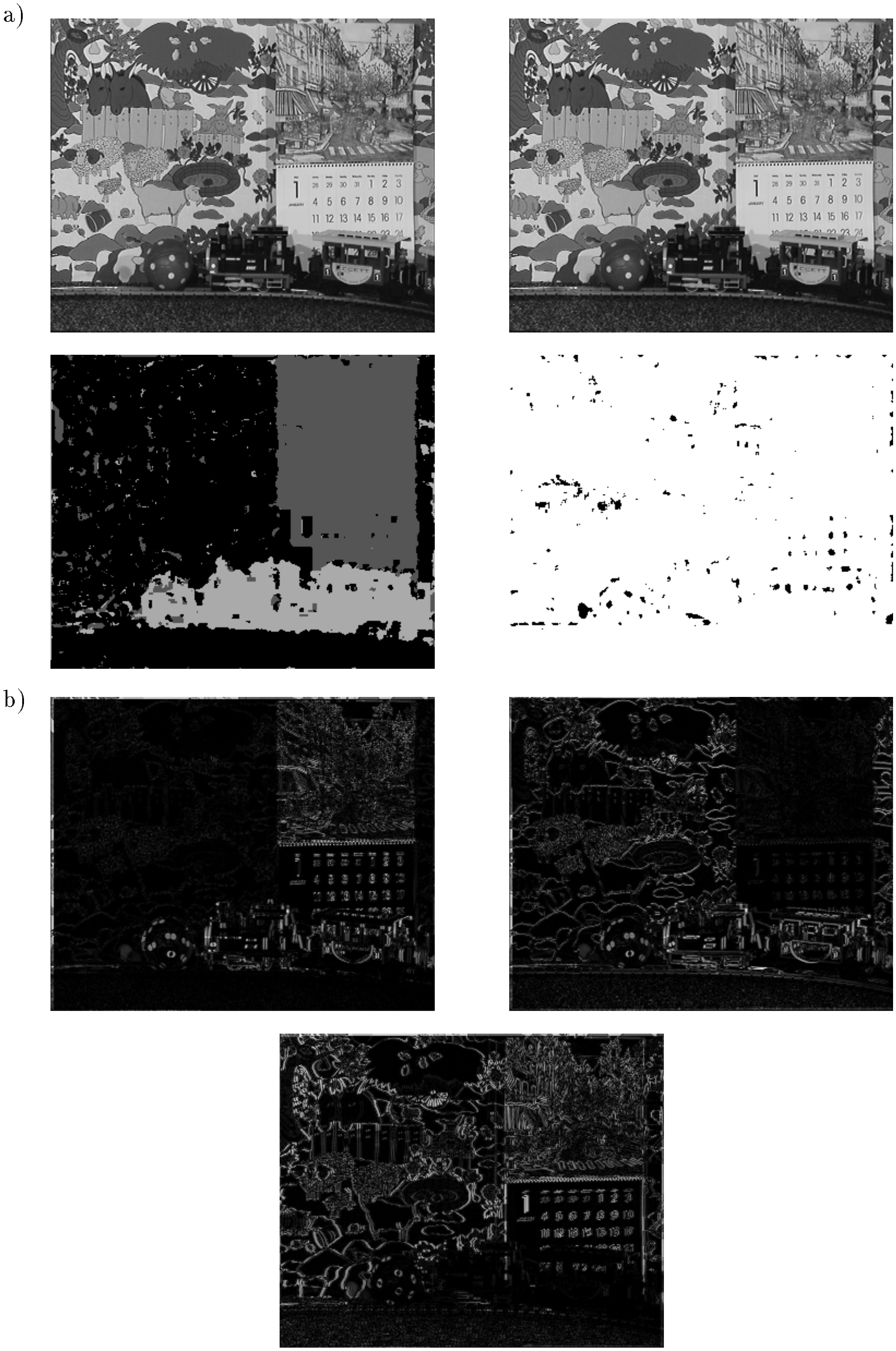

Figure 7: a) Original images, layers of motion and outliers b) Residual errors associated with each layer for the mobile and calendar sequence

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. A practical way to solve for all the parameters is to use the ML estimation for the model and layer ownership parameters using the EM algorithm for Gaussian mixture models given the number of layers. These solutions in turn are used to incrementally test for the right number of models by computing the encoding cost of the model parameters and the resulting data residuals. A highlight of encoding the data cost was the use of the likelihood function of the mixture density. An algorithm that has shown promising results on a variety of sequences has been presented.

A number of issues have yet to be fully explored. First, a multi-frame formulation in which layer ownerships are constrained by a number of frames is required. A formulation for multi-frame motion estimation was presented in [1]. However, the estimation of support layers need to be formulated over multiple frames too. It should also include a mechanism for allowing the number of layers to change over time in order to allow for appearance/disappearance of objects and surfaces.

The viability of using a combination of 2D and 3D models, as appropriate, is also an important issue. Research towards describing image motion with multiple parametric models and the residuals [4], or in terms of a 3D model that combines a 2D parametric and 3D parallax models [12, 20, 22], is in progress. However, creating compact layered descriptions using these representations still needs to be addressed.

In the context of the specific algorithm proposed in this paper, we are further exploring a few issues. First, the use of the non-binary version for describing motion transparency is being experimented with. Especially, the formalism should be extended in order to allow the possibility of using MRF models in the non-binary case also. Second, the relationship of the mixture model to additive, multiplicative and other models of intensity superposition for motion transparency is to be studied more extensively.

Layered representations can serve as useful intermediate representations for recognition and navigation, as well as for video coding and compression. We are actively investigating their usefulness in the context of automatic object and scene representations for image and 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.

A Derivation of the necessary conditions for a maximum of the log-likelihood of mixture of models

The *pdf* of an observation $\mathbf{x}$ as arising from a mixture of models $\{G_1, \ldots, G_g\}$ in some proportions $\pi_1, \ldots, \pi_g$ is given by

$$f(\mathbf{x} \mid \phi) = \sum_{i=1}^{g} \pi_i f_i(\mathbf{x} \mid \mathbf{\Sigma_i}, \boldsymbol{\theta}_i), \qquad \sum_{i=1}^{g} \pi_i = 1, \pi_i \geq 0.$$

The likelihood function for the N independently distributed $\mathbf{x}$'s is

$$\begin{aligned} F(\{\mathbf{X}\} \mid \mathbf{\Phi}) &= \prod_{j=1}^{N} f(\mathbf{x}_j \mid \phi) \\ &= \prod_j \sum_i \pi_i f_i(\mathbf{x}_j \mid \Sigma_i, \boldsymbol{\theta}_i) \end{aligned}$$

Let $\tau_{ij} = Prob(jth$ measurement $\in G_i \mid \mathbf{x}_j, \phi)$. Then,

$$\tau_{ij} = \frac{\pi_i f_i(\mathbf{x}_j|\theta)}{\sum_t \pi_t f_t(\mathbf{x}_j|\theta)} \tag{25}$$

Maximizing the log-likelihood of F, $\max_{\mathbf{\Phi}} \log F(\{\mathbf{X}\} \mid \mathbf{\Phi})$ subject to $\sum_{i=1}^{g} \pi_i = 1$ gives the maximum-likelihood solution to the mixture model problem. A necessary condition for such a minimum to exist is that for the Lagrangian,

$$L_\lambda(\{\mathbf{X}\} \mid \mathbf{\Phi}) = \log F(\{\mathbf{X}\} \mid \mathbf{\Phi}) - \lambda(\sum_{i=1}^{g} \pi_i - 1), \tag{26}$$

the first derivatives with respect to $\mathbf{\Phi}$ and λ should be zero.

Equation (26) can be written as,

$$\begin{aligned} L_\lambda &= \sum_j \log(\sum_i \pi_i f_i(\mathbf{x}_j|\theta)) - \lambda(\sum_{i=1}^{g} \pi_i - 1) \\ &= \sum_j \log \frac{\pi_i f_i(\mathbf{x}_j|\theta)}{\tau_{ij}} - \lambda(\sum_{i=1}^{g} \pi_i - 1) \end{aligned}$$

Differentiating first w.r.t. π_i and setting it to zero, we get,

$$\begin{aligned} \sum_j (\frac{1}{\pi_i} - \frac{1}{\pi_i}(1 - \tau_{ij})) - \lambda &= 0 \\ \sum_j \tau_{ij} &= \lambda \pi_i \end{aligned}$$

Thus, $\sum_i \sum_j \tau_{ij} = \lambda \sum_i \pi_i = \lambda$. Since, by definition, $\sum_i \tau_{ij} = 1$, $\lambda = \sum_j 1 = N$. Therefore,

$$\pi_i = \frac{1}{N} \sum_j \tau ij.$$

Now differentiating L_λ w.r.t. $\boldsymbol{\theta}$, and setting it to zero, we get,

$$\begin{aligned} \sum_j \sum_i \frac{1}{\sum_i \pi_i f_i(\mathbf{x}_j|\theta)} \pi_i \frac{\partial f_i(\mathbf{x}_j \,|\, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= 0 \\ \sum_i \sum_j \tau_{ij} \frac{\partial}{\partial \boldsymbol{\theta}} \log f_i(\mathbf{x}_j \,|\, \boldsymbol{\theta}) &= 0 \end{aligned}$$

after using equation (25) for simplification.

Therefore, the two necessary conditions for a maximum-likelihood estimate of the mixture model parameters are

$$\begin{aligned} \pi_i &= \frac{1}{N} \sum_j \tau ij \\ \sum_i \sum_j \tau_{ij} \frac{\partial}{\partial \boldsymbol{\theta}} \log f_i(\mathbf{x}_j \,|\, \boldsymbol{\theta}) &= 0. \end{aligned}$$

References

[1] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Third European Conference on Computer Vision*, volume 2, pages 316–327, Stockholm, Sweden, May 1994.

[2] J.R. Bergen, P. Anandan, K.J. Hanna, and J. Hingorani. Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision*, pages 237–252, Santa Margherita Ligure, Italy, May 1992.

[3] M.J. Black and P. Anandan. The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. Technical Report TR, Xerox Research Center, Palo Alto, California, December 1993.

[4] M.J. Black and A. Jepson. Estimating multiple independent motions in segmented images using parametric models with local deformations. In *IEEE Workshop on Non-rigid and Articulate Motion*, Austin, Tx, USA, November 1994.

[5] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, pages 173–178, Nassau Inn, Princeton, NJ, October 1991.

[6] C Faloutsos and al. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, July 1994.

[7] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):721–741, November 1984.

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

[9] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representation. In *12th International Conference on Pattern Recognition*, pages 743–746, Jerusalem, Israel, October 1994.

[10] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Second European Conference on Computer Vision*, pages 282–287, Santa Margherita Ligure, Italy, May 1992.

[11] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 760–761, New York, USA, June 1993.

[12] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *12th International Conference on Pattern Recognition*, pages 685–688, Jerusalem, Israel, October 1994.

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

[14] 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, 1986.

[15] 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.

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

[17] 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 Database II.* SPIE, 1994.

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

[19] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection.* John Wiley and Sons, New York, 1987.

[20] H.S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 929–934, Seattle, Washington, USA, June 1994.

[21] G.A.F. Seber and C.J. Wild. *Nonlinear Regression.* John Wiley and Sons, New York, 1989.

[22] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, Washington, USA, June 1994.

[23] J.Y.A. Wang and E.H. Adelson. Layered representation for image sequence coding. In *IEEE International Conference On Acoustics, Speech And Signal Processing*, pages 221–224, Minneapolis, Minnesota, USA, April 1993.

[24] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 361–366, New York, USA, June 1993.