Probabilistic Framework for Feature-Point Matching

CSE6400: Final Report

Ron Tal, MASc Candidate, York University
Supervisory Committee: Minas E. Spetsakis and James Elder
Submitted: June 8, 2009

***Abstract*—In this report we introduce a new method for determining correspondence in a sequence of images. We formulate a probabilistic framework that relates a feature's appearance and its position under relaxed assumptions. We employ a Monte-Carlo approximation for the joint probability density of the feature position and its appearance that uses a flexible noise and motion model to generate random samples. The joint probability density is modeled by a Gaussian Mixture. The feature's position given its appearance is then determined by maximizing its posterior. We evaluate our method using real and synthetic sequences and compare its performance with leading or popular algorithms from the literature. The noise robustness of our algorithm is superior under a wide variety of conditions and offers an effective alternative to hierarchical motion estimation. The method can be applied in the context of optical flow, tracking and any application that needs feature point matching.**

***Index Terms*—Tracking, Optical Flow, Monte-Carlo**

I. Introduction

THE problem of determining feature-point correspondence in multiple images and its variants, like optical flow and contour matching are at the core of many computer vision problems such as motion segmentation, tracking, stereo, structure from motion, etc. The problem of matching in all its forms is ill conditioned and a great variety of assumptions are employed in all the classical solutions [1]. It is usually assumed that neighboring pixels move uniformly from one frame to the other, that image brightness remains constant, that motion is small or that noise is strictly Gaussian. However, these assumptions hold only approximately and we usually treat the effect of their violation as noise. Other sources of noise include both camera and electronic(due to fluctuations of photon and electron arrival), specular distortion and digitization.

To address these issues, several methods have been used to relax one or more of these assumptions. At the core of most such method is a novel formulation that specifically address, often heuristically, a new paradigm regarding one assumption or another. Another common approach is to introduce a more sophisticated noise model that requires the adjustment of many parameters. Since the causes of noise and changes in appearance is sometimes caused by complex physical phenomena, quite often not well understood with mathematically intractable formulations, such methods are not flexible to variations not specifically modeled.

In this work we introduce a probabilistic framework that enables us to incorporate a great variety of noise models. This is achieved by fitting a Gaussian Mixture Model (GMM) on the probability that governs the relationship between feature position and its appearance. To keep the method as general as possible and still be able to employ models that are generative but not analytical, we employ a Monte-Carlo technique [9]. Random samples that represent potential changes of feature point appearance due to displacement and noise are generated about the feature. The Expectation Maximization (EM) clustering algorithm [7] is then used to fit a GMM on the samples, that describes the joint probability of feature position and its appearance. It is of particular importance that our method when applied with the same assumptions used by Lucas & Kanade [11], yields the same algorithm as we show in section III-C. We evaluate our algorithm with experiments on both synthetic image sequences (where noise and motion parameters are known) and real ones.

II. Previous Work

Barron *et al.* provide an excellent performance analysis of the classical solutions to the problem in [1]. The superior performance of the Lucas & Kanade algorithm [11] has made it a popular starting point for further improvement, replacing the Horn & Schunk as the benchmark of choice [17]. Such algorithms relied on a few strong assumptions such as small motion (to enable using a first order Taylor expansion), brightness constancy, smoothness of motion within a small region and no explicit noise model. When combined, such assumptions enables us to formulate the problem using least-squares on an over-determined linear system. Such strong assumptions often do not hold in practice, despite the fact that they seem intuitive at first glance, making them applicable in a very limited set of scenarios or on well conditioned test sequences. In [12] the small motion assumption is addressed using a hierarchical approach, that although it performs well for larger motion, it loses its effectiveness for smaller ones. Black and Anandan [2] use robust statistics [18] and formulate the problem of determining flow using a robust M-Estimator in order to reduce the effect of outliers on accuracy. Negahdaripur [4] has proposed a new definition of optical flow that incorporates both geometric and radiometric cues from the image that addresses violation of the brightness constancy assumption and problems caused by specular reflections. His algorithm had a similar formulation to the one used by Lucas & Kanade with additional unknowns corresponding to affine brightness change model. It successfully relaxes the brightness constancy

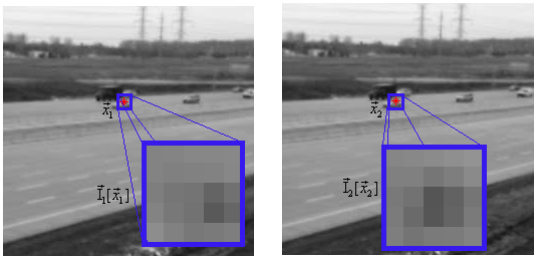
assumption, however, it suffers from the other drawbacks of the Lucas & Kanade algorithm.

Jepson and Black [8] have adapted the EM algorithm for probabilistic mixture models in order to detect multiple motions and thus, handle motion boundaries. Other methods use particle filtering techniques [5]. Wong and Spetsakis [6] provide a noise model that efficiently handles change of brightness and affine deformation of feature appearance. In [13] a RANSAC based algorithm is used for determining multiple motion parameters for the purpose of motion segmentation with the presence of large inter-frame motion. The KLT feature detector [14] can be used to improve tracking by focusing only on highly textured features. The SIFT [15] algorithm can be used to match feature-points in the presence of change of illumination and noise.

III. Our Method

A. In a Nutshell

Many of the classical formulations [11], [17] begin with the assumption that feature appearance does not change across frames. This however is not the case as feature appearance can change in several ways, which can be modeled as stochastic. The intuition behind our approach is that if we compute the joint probability of feature position and its appearance given the image data and assuming a stochastic noise model, we can determine correspondence by finding the maximum likelihood position of the feature in the subsequent frame given its new appearance. In Fig. 1 we show the same image feature in subsequent frames subject to motion and various sources of noise. In our approach we compute this joint probability given



(a) Feature position and its associated appearance at frame 1 (b) Feature position and its associated appearance at frame 2

Fig. 1. Association of feature position and appearance across multiple frames

a parametric noise model using a Monte-Carlo technique. The appearance corresponding feature position in the subsequent frame will maximize the joint probability density function (pdf).

B. The Probabilistic Framework

Given a point $\vec{x}$ on the image plane, we define its *appearance* $\vec{\mathrm{I}}$ to be a vector that comprises all the intensities of the pixels in a small neighborhood around point $\vec{x}$

$$\vec{\mathrm{I}}(\vec{x}) \;=\; \begin{bmatrix} I\left[\vec{x}+\vec{d}_1\right] \\ \vdots \\ I\left[\vec{x}+\vec{d}_k\right] \end{bmatrix}$$

where I is the image as measured by the camera and $\vec{d}_{i=1..k}$ are the positions of the pixels of the neighborhood relative to $\vec{x}$. If two points $\vec{x}_a$ and $\vec{x}_b$ are projections of the same world object on images I_a and I_b respectively, the appearance of the corresponding neighborhoods may change due to camera noise, varying illumination, digitization, deformation etc. so that

$$\vec{\mathrm{I}}_b\left(\vec{x}_b\right) \;=\; \vec{\mathrm{I}}_a\left(\vec{x}_a\right)+\eta \tag{1}$$

where η is noise with pdf $p(\eta \mid \gamma)$ and γ is a set that includes all the parameters of the noise model. In the simplest case the noise model parameters are just a mean and a variance, but in our case will incorporate camera noise, changing illumination, and image jitter and one can easily add other noise components such as affine deformations etc. The conditional probability density of $\vec{\mathrm{I}}_b\left(\vec{x}_b\right)$ can be written as

$$p(\vec{\mathrm{I}}_b\left(\vec{x}_b\right)|\vec{x}_a, I_a, \gamma). \tag{2}$$

This distribution can be determined given the noise model [6] and can then be used, as we show later, to compute

$$p(\vec{x}_a|\vec{\mathrm{I}}_b\left(\vec{x}_b\right), I_a, \vec{x}_b, \gamma) \tag{3}$$

from which the position $\vec{x}_a$ in image I_a that corresponds to the neighborhood $\vec{\mathrm{I}}_b\left(\vec{x}_b\right)$ can be estimated.

The pdf in Eq. (3) expresses the probabilistic model of $\vec{x}_a$ given all that is normally available in a correspondence problem. In such situations we are given neighborhood $\vec{\mathrm{I}}_b\left(\vec{x}_b\right)$ and we try to find the best match in an image I_a. The position of the match, $\vec{x}_a$ can depend on $\vec{\mathrm{I}}_b\left(\vec{x}_b\right)$ and I_a alone but in some cases the pdf $p(\vec{x}_a \mid \vec{x}_b)$ is available as a prior and we can take advantage of it, since we know $\vec{x}_b$. Eq. (3) from Bayes rule becomes

$$p(\vec{x}_a|\vec{\mathrm{I}}_b\left(\vec{x}_b\right), I_a, \vec{x}_b, \gamma) \;= \\ \frac{p(\vec{\mathrm{I}}_b\left(\vec{x}_b\right) \mid \vec{x}_a, I_a, \vec{x}_b, \gamma) p(\vec{x}_a \mid I_a, \vec{x}_b, \gamma)}{p(\vec{\mathrm{I}}_b\left(\vec{x}_b\right) \mid I_a, \vec{x}_b, \gamma)}. \tag{4}$$

In (4) the denominator $p(\vec{\mathrm{I}}_b\left(\vec{x}_b\right) \mid I_a, \vec{x}_b, \gamma)$ is constant with respect to the maximization variable $\vec{x}_a$ and can be ignored. The numeretor can be used in the form that appears in Eq. (4), or can be seen as the joint probability of position and appearance

$$p(\vec{\mathrm{I}}_b\left(\vec{x}_b\right) \mid \vec{x}_a, I_a, \vec{x}_b, \gamma) p(\vec{x}_a \mid I_a, \vec{x}_b, \gamma) \;= \\ p(\vec{x}_a, \vec{\mathrm{I}}_b(\vec{x}_b) \mid I_a, \vec{x}_b, \gamma).$$

Now, given the appearance vector $\vec{\mathrm{I}}_b\left(\vec{x}_b\right)$, we can estimate the corresponding position in image I_a by maximizing the joint probability of image appearance

$$\hat{\vec{x}}_a \;=\; \max_{\vec{x}_a}\left(p(\vec{x}_a, \vec{\mathrm{I}}_b(\vec{x}_b) \mid I_a, \vec{x}_b, \gamma)\right). \tag{5}$$

This solution can form the basis for algorithms for feature point matching, tracking and optical flow as follows. Using an initial image data I_n centered on a feature point $\vec{x}_n$ and a noise and motion model, we first compute a distribution that relates a posible position of the feature given changes in appearance in between frames $p(\vec{x}_{n+1}, \vec{\mathrm{I}}_n(\vec{x}_n) \mid I_{n+1}, \vec{x}_n, \gamma))$. Given a vector $\vec{\mathrm{I}}_{n+1}$ representing a neighborhood in a different image,

we can determine its most likely position by maximizing

$$\vec{x}_{n+1} \;=\; \max_{\vec{x}_{n+1}} \Big(p(\vec{x}_{n+1}, \vec{\mathrm{I}}_n(\vec{x}_n) \mid I_{n+1}, \vec{x}_n, \gamma) \Big)$$

This expression provides us with a powerful framework for the estimation of correspondence of two points in an image sequence under various conditions of noise and motion.

C. Revisiting Lucas & Kanade

To demonstrate the generality and power of this formulation we derive from it the Lucas & Kanade algorithm by introducing its underlying assumptions into the probabilistic framework. The Lucas & Kanade algorithm is perhaps the most successful of the early differential methods for computing optical flow. While it remained obscure for over a decade, a comparative study by Barron, Fleet and Beauchemin [1] showed that it is far superior to the Horn & Schunk algorithm [17] and replaced it as the default benchmark. The algorithm is derived around three core assumptions: brightness constancy, such that feature appearance does not change due to motion; small motion, such that a first-order Taylor series can be used for approximation; and smoothness of flow, which is used to turn an ill-posed problem into an over-determined least-of-squares problem by assuming that motion is identical for every pixel within a small neighborhood. The solution is in the form

$$\vec{\mathbf{u}}_n \;= \mathrm{M}'^{-1}\vec{b} \tag{6}$$

where

$$\mathrm{M}' \;=\; \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix},$$

$\vec{\mathbf{u}}_n$ is the vector of motion parameters at time *n*,

$$\vec{b} \;=\; \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix},$$

I_x and I_y are the directional derivatives of the image, I_t is the time derivative of the image and the summation is over the small neighborhood of feature. In reality, M' may be singular and thus non-invertible. In order to have stable behavior, in most practical implementations of the algorithm Eq. (6) is modified to include a stabilization constant

$$\vec{\mathbf{u}}_n \;= (\mathrm{M}' \;+\; \varepsilon \mathbb{1})^{-1}\vec{b}$$

where ε is usually a small number.

We show later in this section how we can relax these assumptions to obtain a more general solution. The joint probability of feature position and its appearance is implicitly assumed Gaussian in the original paper [11]

$$p(\vec{x}_{n+1}, \vec{\mathrm{I}}_n \mid I_{n+1}, \vec{x}_n, \gamma) \;=\; G(\vec{x}_{n+1}, \vec{\mathrm{I}}_n; \vec{\mu}_{\vec{x},\vec{\mathrm{I}}}, \mathbf{C}_{\vec{x},\vec{\mathrm{I}}}) \tag{7}$$

thus, Eq. (5) can find the value for $\vec{x}_{n+1}$ by minimizing the Mahalanobis distance

$$M_{xI} \;=\; \begin{bmatrix} \vec{\mathrm{I}}_n \;-\; \vec{\mu}_I \\ \vec{x}_{n+1} \;-\; \vec{\mu}_{\vec{x}} \end{bmatrix}^T \mathbf{C}^{-1}_{\vec{x},\vec{\mathrm{I}}} \begin{bmatrix} \vec{\mathrm{I}}_n \;-\; \vec{\mu}_I \\ \vec{x}_{n+1} \;-\; \vec{\mu}_{\vec{x}} \end{bmatrix} \tag{8}$$

where

$$\vec{\mu}_I \;=\; \begin{bmatrix} I_{n+1}\left[\vec{\mu}_{\vec{x}} + \vec{d}_I\right] \\ \vdots \\ I_{n+1}\left[\vec{\mu}_{\vec{x}} + \vec{d}_k\right] \end{bmatrix} = \vec{\mathrm{I}}_{n+1}.$$

Unless we have a prior for $\vec{x}_{n+1}$

$$\vec{\mu}_{\vec{x}} \;=\; \vec{x}_n,$$

vector $\vec{\mathrm{I}}_n$ is the image data in the neighborhood of the feature point at time n and vector $\vec{\mathrm{I}}_{n+1}$ is the image data in the neighborhood of the same feature at time $n+1$. The joint mean of $\vec{x}_{n+1}$ and $\vec{\mathrm{I}}_n$ is

$$\vec{\mu}_{\vec{x},\vec{\mathrm{I}}} \;=\; \begin{bmatrix} \vec{\mu}_I \\ \hdashline \vec{\mu}_{\vec{x}} \end{bmatrix}.$$

The joint covariance matrix of $\vec{x}_{n+1}$ and $\vec{\mathrm{I}}_n$

$$\mathbf{C}_{\vec{x},\vec{\mathrm{I}}} \;=\; E\left\{ \begin{bmatrix} \vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \\ \vec{x}_{n+1} \;-\; \vec{x}_n \end{bmatrix} \begin{bmatrix} \vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \\ \vec{x}_{n+1} \;-\; \vec{x}_n \end{bmatrix}^T \right\}$$

can be partitioned in the following manner

$$\mathbf{C}_{\vec{x},\vec{\mathrm{I}}} \;=\; \left[\begin{array}{c:c} \mathbf{C}_{II} & \mathbf{C}_{Ix}^T \\ \hdashline \mathbf{C}_{xI} & \mathbf{C}_{xx} \end{array}\right]$$

where

$$\mathbf{C}_{II} \;=\; E\Big\{ \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big) \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big)^T \Big\} \tag{9}$$

$$\mathbf{C}_{xx} \;=\; E\Big\{ \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big) \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big)^T \Big\} \tag{10}$$

$$\mathbf{C}_{xI} \;=\; E\Big\{ \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big) \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big)^T \Big\} \tag{11}$$

and the matrix can be efficiently inverted using the method of inversion by partitioning [9]

$$\mathbf{C}^{-1}_{\vec{x},\vec{\mathrm{I}}} \;=\; \left[\begin{array}{c:c} \mathbf{S}_{II} & \mathbf{S}_{xI}^T \\ \hdashline \mathbf{S}_{xI} & \mathbf{S}_{xx} \end{array}\right] \tag{12}$$

where

$$\mathbf{S}_{II} \;=\; \big(\mathbf{C}_{II} \;-\; \mathbf{C}_{Ix}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xI} \big)^{-1} \tag{13}$$

$$\mathbf{S}_{xI} \;=\; -\big(\mathbf{C}_{xx} \;-\; \mathbf{C}_{xI}\mathbf{C}_{II}^{-1}\mathbf{C}_{Ix} \big)^{-1} \big(\mathbf{C}_{xI}\mathbf{C}_{II}^{-1} \big) \tag{14}$$

$$\mathbf{S}_{xx} \;=\; \big(\mathbf{C}_{xx} \;-\; \mathbf{C}_{xI}\mathbf{C}_{II}^{-1}\mathbf{C}_{Ix} \big)^{-1} \tag{15}$$

Using (12) M_{xI} becomes

$$\begin{aligned} M_{xI} \;=\; & \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big)^T \mathbf{S}_{II} \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big) \;+ \\ & \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big)^T \mathbf{S}_{xI} \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big) \;+ \\ & \big(\vec{\mathrm{I}}_n \;-\; \vec{\mathrm{I}}_{n+1} \big)^T \mathbf{S}_{xI}^T \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big) \;+ \\ & \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big)^T \mathbf{S}_{xx} \big(\vec{x}_{n+1} \;-\; \vec{x}_n \big) \end{aligned}$$

We can find the value for $\vec{x}_{n+1}$ that minimizes M_{xI} by taking

partial derivative with respect to $\vec{x}_{n+1}$ and equating it to zero

$$\begin{aligned}\frac{\partial M_{xI}}{\partial \vec{x}} &= 2\left(\ \vec{\mathrm{I}}_n\ -\ \vec{\mathrm{I}}_{n+1}\ \right)\mathbf{S}_{xI}^T \\ &+ 2\left(\ \vec{x}_{n+1}\ -\ \vec{x}_n\ \right)^T\mathbf{S}_{xx} \\ &= 0\end{aligned}$$

from where we get

$$\vec{x}_{n+1}\ =\ \vec{x}_n\ -\ \mathbf{S}_{xI}^T\mathbf{S}_{xx}^{-1}\left(\vec{\mathrm{I}}_n\ -\ \vec{\mathrm{I}}_{n+1}\right)$$

substituting (14) and (15)

$$\vec{x}_{n+1}\ =\ \vec{x}_n\ +\ \mathbf{C}_{xI}\mathbf{C}_{II}^{-1}\left(\vec{\mathrm{I}}_n\ -\ \vec{\mathrm{I}}_{n+1}\right). \tag{16}$$

The noise in the image measurements in [11] can be modeled by

$$\vec{\mathrm{I}}_n(\vec{x}_n) = \vec{\mathrm{I}}_{n+1}(\vec{x}_{n+1}) + \eta_n \tag{17}$$

where η_n is implicitly assumed to be an independent, identically distributed (i.i.d) additive noise. Assuming small motion [11], the variance of $\vec{x}_{n+1}$ is simply $\mathbf{C}_{xx}\ =\ \sigma_{xx}^2\mathbb{1}$ and we can approximate $\vec{\mathrm{I}}_n$ using a first order Taylor Series

$$\vec{\mathrm{I}}_n(\vec{x}_{n+1})\ =\ \vec{\mathrm{I}}_n(\vec{x}_n)\ +\ \nabla\vec{\mathrm{I}}_n^{\ T}(\vec{x}_{n+1}\ -\ \vec{x}_n) \tag{18}$$

where

$$\nabla\vec{\mathrm{I}}_n\ =\ \begin{bmatrix} I_{n,x}\left[\vec{x}_n+\vec{d}_l\right] & I_{n,y}\left[\vec{x}_n+\vec{d}_l\right] \\ \vdots & \vdots \\ I_{n,x}\left[\vec{x}_n+\vec{d}_k\right] & I_{n,y}\left[\vec{x}_n+\vec{d}_k\right] \end{bmatrix}.$$

Now, we can apply (18) and (17) to (11) and (9)

$$\mathbf{C}_{xI}\ =\ \sigma_{xx}^2\nabla\vec{\mathrm{I}}_n^{\ T} \tag{19}$$

and

$$\mathbf{C}_{II}\ =\ \sigma_{xx}^2\nabla\vec{\mathrm{I}}_n\nabla\vec{\mathrm{I}}_n^{\ T}\ +\ \sigma_{\eta\eta}^2\mathbb{1} \tag{20}$$

We invert $\mathbf{C}_{II}$ by applying the Woodbury identity [9] to (20)

$$\mathbf{C}_{II}\ =\ \sigma_{\eta\eta}^2(\mathbb{1}\ -\ \frac{\sigma_{xx}^2}{\sigma_{\eta\eta}^2}\nabla\vec{\mathrm{I}}_n\Phi\nabla\vec{\mathrm{I}}_n^{\ T})$$

where

$$\Phi = (\mathbb{1}\ +\ \frac{\sigma_{xx}^2}{\sigma_{\eta\eta}^2}\nabla\vec{\mathrm{I}}_n^{\ T}\nabla\vec{\mathrm{I}}_n)^{-1}$$

and from (16), (19) and (20)

$$\vec{x}_{n+1}\ =\ \vec{x}_n\ +\ (\frac{\sigma_{nn}^2}{\sigma_{xx}^2}\mathbb{1}+\mathrm{M}')^{-1}\nabla\vec{\mathrm{I}}_n^{\ T}\left(\vec{\mathrm{I}}_n\ -\ \vec{\mathrm{I}}_{n+1}\right) \tag{21}$$

where

$$\mathrm{M}'\ =\ \nabla\vec{\mathrm{I}}_n^{\ T}\nabla\vec{\mathrm{I}}_n\ =\ \begin{bmatrix}\sum I_x^2 & \sum I_xI_y\\ \sum I_xI_y & \sum I_y^2\end{bmatrix}$$

which is the matrix used for the solution proposed by Lucas & Kanade. If we set $\frac{\sigma_{xx}^2}{\sigma_{\eta\eta}^2}=\varepsilon$, then $\mathrm{M}'\ +\ \varepsilon\mathbb{1}=\mathrm{M}$ and we can rewrite (21) as

$$\vec{x}_{n+1}\ =\ \vec{x}_n\ +\ \mathrm{M}^{-1}\nabla\vec{\mathrm{I}}_n^{\ T}\left(\vec{\mathrm{I}}_n\ -\ \vec{\mathrm{I}}_{n+1}\right), \tag{22}$$

and after rearranging (22), we get

$$\vec{x}_{n+1}\ -\ \vec{x}_n\ =\ \mathrm{M}^{-1}\nabla\vec{\mathrm{I}}_n^{\ T}\left(\vec{\mathrm{I}}_n\ -\ \vec{\mathrm{I}}_{n+1}\right). \tag{23}$$

By applying the nomenclature used by Lucas & Kanade, $\vec{x}_{n+1}\ -\ \vec{x}_n\ =\ \vec{\mathbf{u}}_n$ is the motion vector associated with frame n and $\vec{\mathrm{I}}_{n+1}-\vec{\mathrm{I}}_n=I_t$ the time derivative of the image. We now make the observation that:

$$-\nabla\vec{\mathrm{I}}_n^{\ T}\cdot I_t\ =\ \begin{bmatrix}-\sum I_xI_t\\ -\sum I_yI_t\end{bmatrix}=\vec{b}$$

We can now rewrite (23):

$$\vec{\mathbf{u}}_n\ =\mathrm{M}^{-1}\vec{b},$$

which has the same form as (6). An interesting observation to make is that when deriving the Lucas & Kanade formulation by explicitly considering the mathematical implications of their assumptions, the use of a stabilization constant is supported by theory and not by practice only.

Now that we have rederived the solution proposed by Lucas & Kanade using our probabilistic framework, we can relax the Lucas & Kanade assumptions that were incorporated into our framework. In the following sections a more general noise model will be developed, taking into account random fluctuations of light intensity and affine deformation. The assumption of a Gaussian pdf given in (7) will be replaced with a GMM.

D. Modeling Probabilities Using a Mixture of Gaussians

Now that we have introduced our probabilistic framework and used it to derive the Lucas-Kanade correspondence method by incorporating its implicit assumptions, we can extend our method by relaxing the assumptions. Using a GMM in place of a single Gaussian, we can attain a better approximation of any arbitrary joint pdf of image appearance and position. The pdf from (7) can thus be reformulated as:

$$p(\ \vec{x}_{n+1}\ ,\ \ \vec{\mathrm{I}}_n\ \mid I_{n+1},\vec{x}_a,\gamma)\ =\ \sum_{i=1}^{N}\pi_jG(\vec{x},\vec{\mathrm{I}};^j\vec{\mu}_{\vec{x},\vec{\mathrm{I}}},^j\mathbf{C}_{\vec{x},\vec{\mathrm{I}}}) \tag{24}$$

where π_i is the mixture prior, ${}^j\vec{\mu}_{\vec{x},\vec{\mathrm{I}}}$ is the mean and ${}^j\mathbf{C}_{\vec{x},\vec{\mathrm{I}}}$ the covariance matrix of the j^{th} component. There are two distinct problems that we must solve: fitting and maximization. The method that we use to fit a joint pdf for position and image appearance is described later in this section. In the previous section we have shown how we could find the maximum likelihood value for $\vec{x}$ given $\vec{\mathrm{I}}$ simply by maximizing the log-likelihood, when using the Lucas & Kanade assumptions. However, using a GMM this cannot be directly done as we cannot derive a closed form solution for $\vec{x}$. Instead, we first determine the mixture component k that is the most likely cause of our data $\vec{\mathrm{I}}$ by minimizing the Mahalanobis distance as determined in (8) with parameters

$${}^k\mu\ =\ \left[\begin{array}{c}{}^k\vec{\mu}_I\\ \hline {}^k\vec{\mu_x},\end{array}\right]$$

$$ {}^{k}\mathbf{C}_{\vec{x},\vec{\mathrm{I}}} \;=\; \left[\begin{array}{c|c} {}^{k}\mathbf{C}_{II} & {}^{k}\mathbf{C}_{Ix}^{T} \\ \hline {}^{k}\mathbf{C}_{xI} & {}^{k}\mathbf{C}_{xx} \end{array}\right] $$

and $\vec{x}$ is then calculated as in (16):

$$ \vec{x} \;=\; {}^{k}\vec{\mu}_{x} \;+\; {}^{k}\mathbf{C}_{xI}\,{}^{k}\mathbf{C}_{II}^{-1}\left(\vec{\mathrm{I}} \;-\; {}^{k}\vec{\mu}_{I}\right). \qquad (25) $$

Although (25) is only an approximate closed form solution for $\vec{x}$, the empirical evidence we present in Sec. IV-A indicates that the approximation has negligible contribution to the error. Next we introduce the statistical model we use to form (24).

E. Noise Model

The noise model should reflect possible changes to image appearance between two frames. We use a similar noise model to the one utilized by Wong and Spetsakis [6]. Besides inter-frame motion and i.i.d camera noise that were considered previously, we can expand our noise model to contain a wider range of changes and deformations such as random fluctuations of illumination and pixelwise jitter. We model the image appearance at a given time as a function of the appearance at the previous frame:

$$ \begin{aligned} \vec{\mathrm{I}}_{n+1}[\vec{x}_b] \;&=\; \vec{\mathrm{I}}_{n}[\vec{x}_a \;+\; \vec{u}_n] \;+\; \vec{\eta}_n \\ &+\; \mathit{Diag}(\vec{I}_x[\vec{x}_a \;+\; \vec{u}_n])\vec{\varpi}_n \\ &+\; \mathit{Diag}(\vec{I}_y[\vec{x}_a \;+\; \vec{u}_n])\vec{\epsilon}_n \\ &+\; \vec{\mathrm{I}}_{n}[\vec{x}_a \;+\; \vec{u}_n]\beta \;+\; \vec{1}\alpha \end{aligned} $$

where $\mathit{Diag}(\vec{I}_x)$ is a diagonal matrix consisting of the elements of $\vec{I}_x$ and similarly for $\vec{I}_y$. The random variables are: $\vec{\eta}_n$, a random vector representing i.i.d camera noise, $\vec{\varpi}_n$ and $\vec{\epsilon}_n$, also i.i.d random vectors representing small horizontal and vertical pixelwise motion within a neighborhood (like leaves fluttering in the wind), β, a random scalar that reflects multiplicative change of illumination with respect to the original image and α, a random scalar reflecting additive change of illumination as proposed by Negahdaripour [16] that it is normally sufficient to model brightness change as affine. The inter-frame motion of the feature is represented by the flow vector $\vec{u}_n$, which is a random vector that reflects our expectations of feature motion between frames.

We now complete our solution by fitting a GMM for our uncertainty model using a generative approach.

F. Generative Model for Probability Distributions

Now that we have introduced a realistic uncertainty model and an efficient way to determine the most likely position of a feature point, we need a way to fit a pdf that reflects our model. We approximate the GMM using a Monte-Carlo approach: we first take random samples of neighborhoods that reflect changes of feature appearance due to both displacement and noise (as described by the noise model, introduced in the previous section) and then approximating a GMM using the EM clustering algorithm.

The random samples used by the Monte-Carlo method should reflect our assumptions regarding noise and displacement. For each feature that we want to track we take *N* random samples where the formula for the i^{th} sample can be written as

$$ \begin{aligned} P_i \;&=\; \vec{\mathrm{I}}_{n}[\vec{x}_a \;+\; \vec{u}_n] \;+\; \vec{\eta}_n \\ &+\; \mathit{Diag}(\vec{I}_x[\vec{x}_a \;+\; \vec{u}_n])\vec{\varpi}_n \\ &+\; \mathit{Diag}(\vec{I}_y[\vec{x}_a \;+\; \vec{u}_n])\vec{\epsilon}_n \\ &+\; \vec{\mathrm{I}}_{n}[\vec{x}_a \;+\; \vec{u}_n]\beta \;+\; \vec{1}\alpha \end{aligned} $$

where P_i is a possible appearance of the neighborhood around the feature in the subsequent frame. The random variables presented during the discussion of our noise model are used to generate a random observation for each sample. Their distribution parameters can be modified according to a specific application or when certain knowledge about the scene is known. For example, in a stereo problem the random motion $\vec{u}$ can be distributed according to the epipolar constraint. In our tracking examples presented in Sec. IV-B, no prior information regarding the motion of vehicles was assumed, thus $\vec{v}$ was isotropically distributed about the initial feature position with $\sigma = 7\ pix$. The other random variables are generated in accordance of their corresponding parameters in the noise model. For example, ϖ and ϵ are i.i.d vectors generated according to a normal distribution with $\sigma = 0.25$ to represent small subpixel jitter. $\vec{\eta}$ represents camera noise and is thus an i.i.d normal vector with $\sigma = 2\ greylevels$.

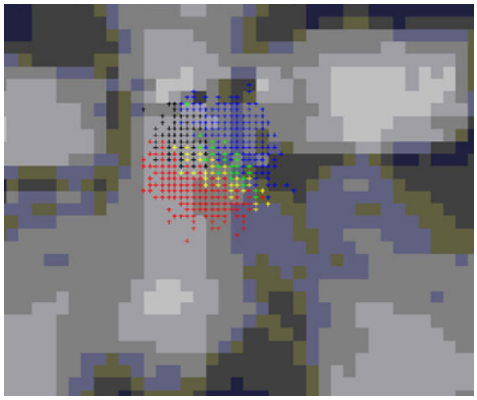


Fig. 2. An illustration of 1000 random samples about the feature point clustered into 5 components in 27 dimensions

The samples that we generate according to the noise model provide us with statistical behavior of feature appearance. After generating a sufficient number of sample points, we find the GMM that best fits the samples using the EM algorithm, where each pixel position relative to the center of every neighborhood sample is a separate dimension. In our tests we have used 5-by-5 neighborhoods, giving us 27 dimensions in total, 25 for appearance and 2 for position. An illustration of clustered random samples generated using our method is presented in Fig. 2.

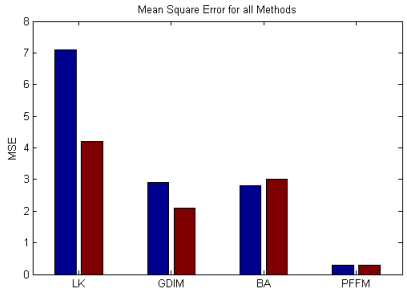
IV. Experimental Results

The results are presented in two parts. First, a quantitative evaluation of our correspondence method under known noise and motion parameters is presented, a long with a comparative study of other methods from the literature using still

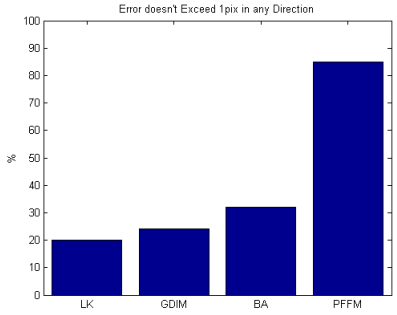
real images with synthetically generated motion and noise. Second, we provide a qualitative demonstration of the tracking algorithm using real world data, with no known ground truth.

A. Synthesized Noise and Motion

To gauge the performance of of our method we use a real image and apply a known noise and motion model to it. We apply three types of modification to image appearance: i.i.d white camera noise with $\sigma = 2$ $greylevels$, random change of brightness $\sigma = \%5$, and random motion, which is isotropically distributed in the range of $0 \ldots 3$ about the initial feature position. Subpixel accuracy in motion is attained using a 5 shot polynomial interpolation. We then randomly select good trackable features using the KLT method [14] and compare the Mean Square Error (MSE) measure produced using four correspondence methods: Lucas & Kanade (LK) [11], Negahdaripour's Generalized Dynamic Image Model (GDIM) [16], Black & Anandan's robust flow (BA) [3], and the Probabilistic Framework for Feature-Point Matching we have proposed in this paper (PFFM). To model changes in appearance in the PFFM model, we use the same noise and motion parameters that are used to generate each frame.



(a) Mean Square Error using each of the methods. The blue bars represent MSE in the horizontal direction and the red bars represent errors in the vertical direction.



(b) Overall reliability of each method.

Fig. 3. Comparison of overall performance

As may be expected, having good statistical estimate of changes in image appearance lead to better success in correspondence. A bar chart of the MSE attained using each method, in both the horizontal (blue) and vertical (red) are presented in Fig. 3(a). Using the Lucas & Kanade algorithm in this context, the new feature point position is recovered with a MSE measure of 7.1 for horizontal and 4.2 for vertical. Using the GDIM method, brightness change is recovered, providing better results. Using this method the new feature point position is recovered with a MSE measure of 2.9 for horizontal and 2.1 for vertical. The BA method promises improved accuracy as it robustly fits motion parameters, while eliminating outliers. However, we have found that performance was comparable to that of GDIM with MSE measures of 2.8 in the horizontal direction and 3.0 in the vertical direction. Using the PFFM method we attain the greatest accuracy with MSE measures of 0.3 for both horizontal and vertical. The PFFM is also consistently reliable, with error not exceeding 1 pixel in any direction %86 of the times (see Fig. 3(b)). We have plotted

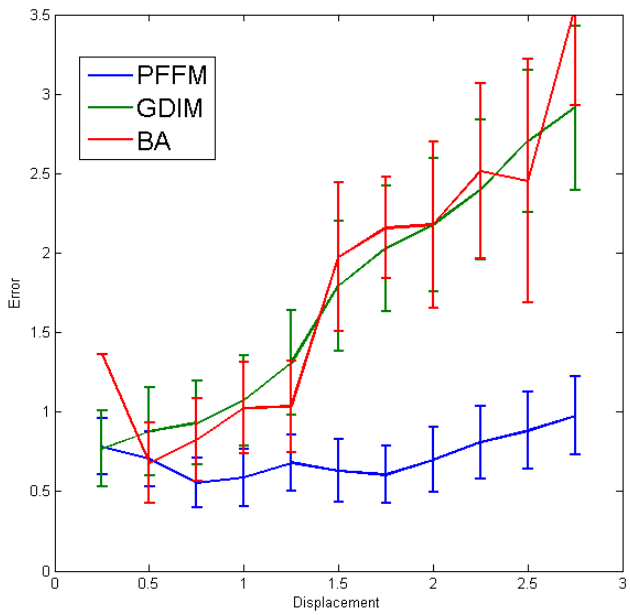


Fig. 4. Comparison of PFFM with GDIM and BA. The graph shows error as a function of displacement. LK is not shown as it barely fits within the desired plot region

error as a function of total displacement for each method in Fig. 4. We have omitted LK from this plot as it was unreliable even for small displacements and would thus not fit in the desired plot size. The graph indicates that our method is more reliable even for larger displacements.

Our method of selecting a component based on the Mahalanobis distance picks the component that produces the minimal error %70 of the time. We have determined that the total contribution to the MSE measure caused by picking the wrong component is 0.04. This was done by computing a penalty equal to the difference between the error resulting from the cluster we have selected and the minimal error produced by any cluster.

B. Tracking Vehicles on a Highway

In order to demonstrate the applicability of our method to tracking, we have acquired a short video of vehicles on a freeway. The sequence was captured using a standard point-and-shoot digital camera's video option at 24 fps. The camera

was hand held and the photographer was purposely shaking the camera during the capture. Image resolution is 640x480 and a typical frame is shown in Fig. 5. We select good

Fig. 5. Frame 30 from the video sequence

(a) Frame 15 with the selected feature in red

(b) Frame 30 with the tracked feature in red

Fig. 6. Tracking a feature point over 15 frames

trackable features using the KLT method [14] and choose to track one that belongs to a moving vehicle. The feature we have selected falls on the front bumper of a sedan in frame 15 (Fig. 7(a)). We have tracked the target for 15 frames and the result is shown in Fig. 7(b). Although the target has drifted a bit backwards to the sedan's window, our method successfully kept up with the motion of the vehicle. During the sequence the vehicle has moved about 100 pixels with single frame motion ranging between negligible and about 15 pixels. Overall, feature appearance as represented by a neighborhood of 5x5 pixels around it is not a strong descriptor and cannot form the basis of a tracking algorithm on its own, as it may drift slightly every frame ultimately loosing the target. Our results indicate that our algorithm can be used as a way to determine correspondence under relaxed assumptions of noise and motion of a more robust tracking algorithm that uses multiple feature points on an object.

Our method does not explicitly model motion boundaries and will thus be susceptible to failure when a tracked object crosses a motion boundary. Another issue with our method is that of efficiency. Depending on the parameters in the noise model, a large number of samples may need to be clustered. The EM algorithm is not time efficient and is currently the bottleneck in our method. For each tracked point our method takes a couple of seconds per frame.

V. Conclusion

We have presented a new algorithm for feature-point matching that uses a probabilistic framework to model uncertainty regarding changes in image appearance. We have further shown that when our probabilistic framework is used in conjunction with the assumptions inherent to the Lucas & Kanade algorithm, we can reduce our solution to theirs. Earlier methods that relied on such strong assumptions can not be easily extended to handle additional causes of deformation without loosing their flexibility. Our probabilistic framework is flexible in the sense that a great variety of noise models can be easily incorporated. Since we do not have a closed form joint probability density function of feature position and its appearance, we approximate it using a generative approach. We use a Monte-Carlo technique to generate random samples of possible appearance of image data under our noise model and fit a GMM using EM clustering. Using synthetic image sequences that simulate relaxed assumptions, we have demonstrated that our method performs better than some popular two of the best algorithms in the literature. Our comparative study included the Lucas & Kanade algorithm, Negahdaripour's Generalized Dynamic Image Model and Black & Anandan robust flow. We have demonstrated the efficacy of our algorithm in feature-point tracking using sequences of freeway traffic. Our algorithm has performed generally well when tracking good features at the absence of motion boundaries. In future work we will extend our tracking algorithm, using an affine model on multiple feature points in order to attain more robust performance.

References

[1] J. L. Barron, D. J. Fleet and S. S. Beauchemin, *Performance of Optical Flow Techniques*, International Journal of Computer Vision, 12(1):43-77, 1994.

[2] Michael. J. Black and P. Anandan, *A Framework for the Robust Estimation of Optical Flow*, Fourth International Conference On Computer Vision, 231-236, 1993.

[3] Michael J. Black and P. Anandan, *The Robust Estimation of Multiple Motions: Parametric Piecewise-Smooth Flow Fields*, Computer Vision and Image Understanding, 63(1):75-104, 1996.

[4] S. Negahdaripour, *Revised Representation of Optical Flow for Dynamic Scene Analysis*, Proc. ISCV, Coral Gables, Fla., 1995.

(a) Frame 12 showing a small tracked vehicle on the off-ramp nearing a hedge. The tracked feature is located at the bottom of the vehicle in between the axels. This feature-point is trackable when unobstructed.

(b) Frame 15 showing tracker failure at a motion boundary when target becomes partially occluded. The tracked feature is being obstructed by the hedge after frame 13. The tracker keeps up with the vehicle until the entire axel crosses the hedge at frame 15 at which point the target is lost.

Fig. 7. A sequence showing tracker failure at a motion boundary

[5] Michael J. Black and David. J. Fleet, *Probabilistic Detection and Tracking of Motion Boundaries*, International Journal of Computer Vision, 38(3):231-245, 2000.

[6] King Yuen Wong and Minas E. Spetsakis, *Motion Segmentation by EM Clustering of Good Features*, Conference on Computer Vision and Pattern Recognition Workshops, 166-166, 2004.

[7] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society Series B, 39(1):1-38, 1997.

[8] A. Jepson and Michael. J. Black, *Mixture Models for Optical Flow Computation*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 760-761, 1993.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[10] A. D. Jepson, D. J. Fleet and T. F. El-Maraghi, *Robust Online Appearance Models for Visual Tracking*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10):1296-1311, 2003.

[11] B. Lucas and T. Kanade, *An Iterative Image Registration Technique with an Application in Stereo Vision*, Proc. DARPA IU Workshop, 121-130, 1981.

[12] J. R. Bergen, P. Anandan, K. J. Hanna and R. Hingorani, *Hierarchical Model-Based Motion Estimation*, Proceedings of the Second European Conference on Computer Vision, 237-252, 1992.

[13] J. Wills, S. Agarwal, S. Belongie, *What Went Where*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 37-44, 2003.

[14] J. Shi, C. Tomasi, *Good Features to Track*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 593-600, 1994.

[15] David G. Lowe, *Object Recognition from Local Scale-invariant Features*, Proceedings of the International Conference on Computer Vision 2, 1150-1157, 1999.

[16] S. Negahdaripour, *Revised Definition of Optical Flow: Integration of Radiometric and Geometric Cues for Dynamic Scene Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence,20(9):961-979, 1998.

[17] B. K.P Horn and B. G. Schunk, *Determining Optical Flow*, Proc. Artificial Intelligence, vol. 17, 1981.

[18] Peter J. Huber, *Robust Statistics*, Wiley Series in Probability and Mathematical Statistics, 1981.