Probabilistic Data Association Methods for Tracking Complex Visual Objects

Christopher Rasmussen and Gregory D. Hager, *Member*, *IEEE*

Abstract—We describe a framework that explicitly reasons about data association to improve tracking performance in many difficult visual environments. A hierarchy of tracking strategies results from ascribing ambiguous or missing data to: 1) noise-like visual occurrences, 2) persistent, known scene elements (i.e., other tracked objects), or 3) persistent, unknown scene elements. First, we introduce a randomized tracking algorithm adapted from an existing probabilistic data association filter (PDAF) that is resistant to clutter and follows agile motion. The algorithm is applied to three different tracking modalities—homogeneous regions, textured regions, and snakes—and extensibly defined for straightforward inclusion of other methods. Second, we add the capacity to track multiple objects by adapting to vision a joint PDAF which oversees correspondence choices between same-modality trackers and image features. We then derive a related technique that allows mixed tracker modalities and handles object overlaps robustly. Finally, we represent complex objects as conjunctions of cues that are diverse both geometrically (e.g., parts) and qualitatively (e.g., attributes). Rigid and hinge constraints between part trackers and multiple descriptive attributes for individual parts render the whole object more distinctive, reducing susceptibility to mistracking. Results are given for diverse objects such as people, microscopic cells, and chess pieces.

Index Terms—Visual tracking, data association, color regions, textured regions, snakes.

1 Introduction

Traditionally, the emphasis in framing the visual tracking problem has been on *estimation* [1], [2]. Given a sequence of images containing the object that we would like to represent concisely with a parametric model, an *estimator* is a procedure for finding the parameters of the model which best fit the data. Most of the image data is typically irrelevant, so, if the object's image projection can be unambiguously discriminated from the rest of the image, it is segmented and used exclusively for estimation.

Under real world conditions, it can be difficult to accurately identify an object's image projection because visual phenomena such as agile motion, distractions, and occlusions interfere with estimation. We define *agile motion* as a sustained object movement that exceeds a tracker's dynamic prediction abilities. Its occurrence undermines the estimation process because it renders the putative location of the object's image projection uncertain, complicating efficient segmentation. A further obstacle to clear-cut segmentation is a *distraction* or another scene element which has a similar image appearance to the object being tracked. Finally, *occlusion* results when another scene element is interposed between the camera and the tracked object, blocking a portion of the object's image projection. This results in incomplete data or no data being supplied to the estimation algorithm.

- *C. Rasmussen is with the National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899. E-mail: crasmuss@nist.gov.*
- *G.D. Hager is with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218. E-mail: hager@cs.jhu.edu.*

Manuscript received 28 Oct. 1999; revised 17 Aug. 2000; accepted 29 Jan. 2001.
Recommended for acceptance by M.J. Black.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110856.

We tackle these problems with two broad approaches. First, we adapt to vision several existing *data association* [3] versions of the Kalman filter [4] constructed to handle certain classes of these occurrences and make novel improvements to them. The second part of our strategy is a method of defining a tracked object more *distinctively* so that visual disruptions happen less frequently and with less severity. As our algorithms are based on the Kalman filter, they work with point-like *measurements* rather than directly on images. Another major component of this paper is therefore a process for segmenting and summarizing a discrete set of image areas that resemble the target (where the similarity metric depends on the modality used for tracking). Thus, the term "measurement" serves as a convenient shorthand for coherent subsets of the image data that may be used for state estimation and data association serves to weight the influence of these alternatives.

In the next section, we review the probabilistic foundations of the visual tracking problem. In Section 3, we derive a formulation of image similarity for three modalities that rely on color, shape, or appearance to define the target. Section 4 analyzes the image preprocessing necessary to adapt data association filters to tracking a single object visually. Section 5 examines the problem of interference caused by other known objects. Section 6 introduces methods for describing a tracked object more distinctively in order to minimize the deleterious effects of unknown, persistent distractions in the scene. Results for all of these methods are presented in Section 7. We survey related work on tracking in Section 8 and sum up our contributions in Section 9. A table summarizing the key steps of each algorithm is given in the Appendix (Fig. 16).

2 Background

Mumford [5] and others have suggested that many problems in vision may be cast as an attempt to find a *maximum a posteriori* (MAP) estimate [1] of the state of the world given a signal that is a transformed version of it. Bayes' theorem [5], [6] provides a tool for reasoning probabilistically about the world $\mathbf{W}$ from the image seen $\mathbf{I}$: $p(\mathbf{W} \mid \mathbf{I}) = \frac{p(\mathbf{I}|\mathbf{W})p(\mathbf{W})}{p(\mathbf{I})}$. $p(\mathbf{I})$ can be deduced from the other terms, so it is typically treated as a normalizing constant $1/k$. A MAP estimate of the state of the world (not necessarily unique) is a maximally likely one given the observed image: $\operatorname{argmax}_{\mathbf{w}} \mathrm{p}(\mathbf{W} \mid \mathbf{I})$. To track, an observer focuses its interest on a small part of the world, which we call an *object* or *target* and takes past images into account. At time t, the state $\mathbf{X}_t \in \mathcal{X}$ represents the current estimate of the object's salient parameters. Using the sequence of images $\mathbf{I}_t, \mathbf{I}_{t-1}, \ldots$ observed so far, the MAP tracking task is to estimate a state that maximizes $p(\mathbf{X}_t \mid \mathbf{I}_t, \mathbf{I}_{t-1}, \ldots)$. Applying Bayes' theorem and rearranging yields the following expression [3], [7]:

$$p(\mathbf{X}_t \mid \mathbf{I}_t, \mathbf{I}_{t-1}, \ldots) = k_t p(\mathbf{I}_t \mid \mathbf{X}_t) p(\mathbf{X}_t \mid \mathbf{I}_{t-1}, \mathbf{I}_{t-2}, \ldots). \quad (1)$$

Here, $p(\mathbf{X}_t \mid \mathbf{I}_{t-1}, \mathbf{I}_{t-2}, \ldots)$, which summarizes prior knowledge about $\mathbf{X}_t$, is a prediction based on the previous state estimate and knowledge of the object's dynamics. Asserting that object dynamics are such that states form a Markov chain [6] obtains

$$p(\mathbf{X}_t \mid \mathbf{I}_{t-1}, \mathbf{I}_{t-2}, \ldots) = \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t \mid \mathbf{X}_{t-1}) p(\mathbf{X}_{t-1} \mid \mathbf{I}_{t-1}, \mathbf{I}_{t-2}, \ldots).$$

Dropping time indices for clarity, $p(\mathbf{X} \mid \mathbf{I})$ describes the probability of observing a particular image at time t given the current state. We call this the *image likelihood*. The image likelihood depends on the physics of image formation and noise that may corrupt what is expected [8]. Let the space of images be $\mathcal{I}$ and $\pi : \mathcal{X} \to \mathcal{I}$ be an *image prediction* function describing the expected image projection of the target given a particular state. If they are not explicitly included in $\mathbf{X}$, assumptions must be made in π about lighting, occlusions, background, object reflectance properties, camera variables such as focal length, etc.

An efficient algorithm for computing the MAP estimate of (1) when $p(\mathbf{X}_t \mid \mathbf{I}_t, \mathbf{I}_{t-1}, \ldots)$ is Gaussian is the Kalman filter [3], [4] (see the Appendix for details). In order for $p(\mathbf{X}_t \mid \mathbf{I}_t, \mathbf{I}_{t-1}, \ldots)$ to be Gaussian, $p(\mathbf{I} \mid \mathbf{X})$, $p(\mathbf{X}_t \mid \mathbf{X}_{t-1})$, and the prior probability of the state before any images are viewed must be Gaussian. Some possible causes of and remedies for non-normality are discussed in [7]; in Sections 4 and 5, we present data association filters that handle certain kinds of violations of the Kalman filter's assumptions. In nonvision tracking domains such as radar [3], measurement extraction as a precursor to applying the Kalman filter is fairly simple. A target might be simply a bright point on a dark background, so thresholding alone quickly segments out high-likelihood hypotheses for the target location. Generating visual target measurements, however, is usually more difficult than thresholding and requires more information than just image location. Possible measurement parameters include geometric characteristics such as the location of the area's center and its height, width, and orientation. These parameters define a *measurement space* $\mathcal{Z}$ such that a point $\mathbf{Z} \in \mathcal{Z}$ is related to a state $\mathbf{X}$ via a continuous *measurement function* $H(\mathbf{X}) = \mathbf{Z}$. The measurement function may simply reduce the dimensionality of $\mathbf{X}$ by dropping its temporal parameters or describe a more complicated relationship between what is measured and what is estimated.

The bases for $p(\mathbf{I} \mid \mathbf{X})$ and, therefore, for the measurement generation procedure described in Sections 4 and 5, are the form of the predicted target image projection $\pi(\mathbf{X})$ and the method for quantifying the similarity of the image $\mathbf{I}$ to that prediction. Both of these depend on what we call the *modality* used to identify the object. A modality is a visual attribute such as shape, color, direction of motion, etc., that might constitute a tracking algorithm's complete description of its target. For example, suppose we want to track a bright red ball. We might choose a color modality to predict the hue of the ball's circular image projection and to define a metric on circular areas of hue in order to gauge the similarity of our prediction to the actual image. This method does not exploit all available image information about the ball (ignoring, for instance, any designs printed on it or its motion), but makes a choice about what information is relevant and adequate.

3 Tracking Modalities

This section covers the form of the likelihood function $p(\mathbf{I} \mid \mathbf{X})$ for three modalities used to analyze the image: homogeneous regions, textured regions, and snakes.

3.1 Homogeneous Regions

We define a *region* as the image projection of a simply connected *patch* of a smooth surface. Let c_P be a function describing the intrinsic color pattern over a patch P in RGB space (akin to a computer graphics texture map [9]). If $c_P(P)$ is roughly constant, then we call region R a *homogeneous* region. In previous work [10], [11], we described a method based on the Dichromatic Reflection Model [12] for modeling a given region $R's$ color by having the user manually select a set of pixels in R that are nonhighlighted, nonsaturated, and have significant intensity variation. Using singular value decomposition [2], an ellipsoid parametrized by a translation matrix $\mathbf{T}$, a rotation matrix $\mathbf{R}$, and a scale matrix $\mathbf{S}$ is fit to the sampled pixels' color distribution in RGB space. The Mahalanobis distance [13] $\gamma(\mathbf{I}(x,y), \mathbf{T}) = |\mathbf{S}^{-1}\mathbf{R}^T(\mathbf{I}(x,y) - \mathbf{T})|$ is used to measure the similarity γ between the predicted pixel color $\mathbf{T}$ and the actual color $\mathbf{I}(x,y)$ at each pixel (x,y).

Color information is combined with a geometric representation of R as a rectangle parametrized by image position x, y, size w, h, and orientation ϕ. The rectangle C used to represent R is the best-fitting one according to an objective function f that is minimized by minimizing the sum of γ over all pixels inside C while maximizing it outside. The local image neighborhood of the *positive center* C is delineated by a rectangular border F which we call the *inhibitory frame*. To balance its influence on f, F is sized so that $|F| = |C|$ while maintaining the same aspect ratio. A

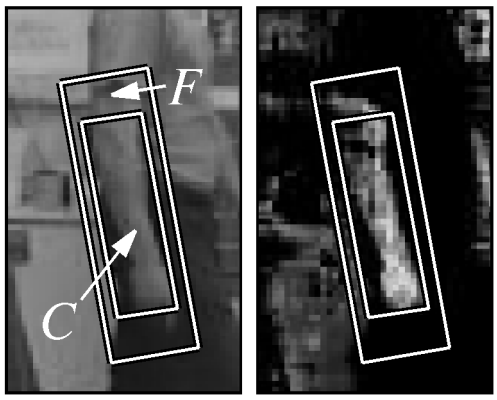


Fig. 1. Homogeneous region. (a) The geometry of a region for an arm tracker with the positive center C and inhibitory frame F labeled. (b) Pixel similarity γ of the image to the modeled arm skin color, with R's geometry overlaid. (Input image courtesy of J. MacCormick).

good but suboptimal fit of a rectangle and its frame to a human arm region is shown in Fig. 1.

These conditions are satisfied by the following expression for the image likelihood of a homogeneous region:

$$p_{hregion}(\mathbf{I} \mid \mathbf{X}) = \mathrm{sig}\left(\frac{1}{\sigma^2_{hregion}} \sum_{x,y \in C \cup F} a(x,y) \cdot \psi_{hregion}(x,y)\right), \tag{2}$$

where $\mathrm{sig}(x) = \frac{1}{1+e^{-x}}$ and $a(x,y)$ is the fraction of the total area $|R|$ of the region R represented by the pixel at (x,y). The degree to which each pixel in the region fits the membership model is given by:

$$\psi_{hregion}(x,y) = \begin{cases} -\gamma\big(\mathbf{I}(x,y), \mathbf{T}\big) & \text{if } (x,y) \in C \\ \gamma\big(\mathbf{I}(x,y), \mathbf{T}\big) & \text{if } (x,y) \in F. \end{cases} \tag{3}$$

3.2 Textured Regions

A *textured* region is defined as a region whose patch P has an intrinsic color pattern $c_P(P)$ with significant vertical and horizontal intensity gradients. This allows sum-of-squared-differences (SSD) methods [14], [15], [16] to successfully estimate the region's geometric and photometric transformations. Here, we limit our attention to affine geometric transformations of an intensity patch whose projection is approximated by a rectangle.

We write $c_R(R)$ to denote the pattern by which a textured region R is recognized. It is modeled by a user-selected rectangular image sample $\mathbf{I}_R$ of the target called the *reference* image. An example of the selection step is shown in Fig. 2a and the resulting reference image in Fig. 2c. During tracking, the object state $\mathbf{X}$ specifies the shape of R as an affine warp $\mathbf{A}$ of the reference image, yielding a predicted image $\mathbf{I}_P$. In practice, the image inside the rectangle predicted by $\mathbf{X}$ is inversely warped using $\mathbf{A}^{-1}$ with bilinear interpolation [9] to get a comparison image $\mathbf{I}_C$ that is the same size as the reference image. An example of the predicted shape and location for the textured region referred to above is shown in Fig. 2b; its associated comparison image is depicted in Fig. 2d.

The gradient of textured regions makes feature comparison within regions sufficient to measure scaling, obviating the inhibitory frame necessary for homogeneous regions. An SSD formulation expresses the image likelihood as inversely proportional to the difference between the reference image and the comparison image:

$$p_{tregion}(\mathbf{I} \mid \mathbf{X} = \exp\left(-\frac{1}{\sigma^2_{tregion}} \sum_{x,y \in \mathbf{I}_R} a(x,y) \cdot \psi_{tregion}(x,y)\right), \tag{4}$$

where $a(x,y)$ is the fraction of $|\mathbf{I}_R|$ represented by the pixel at (x,y) and

$$\psi_{tregion}(x,y) = \big(\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y)\big)^2. \tag{5}$$

An image representing the residual for the example is shown in Fig. 2e.

3.3 Snakes

We define a *snake* [17], [18] as the projection of a continuous *contour* lying on a smooth surface onto the image. The contour may delineate a contrast edge, the surface silhouette, or a simple line; in this paper, we assume that the contrast takes the form of an intensity difference, permitting the use of standard edge detection algorithms. We have found that the Canny algorithm [19] gives excellent results, though the Sobel edge operator [20], while somewhat less sophisticated, gives adequate results more speedily.

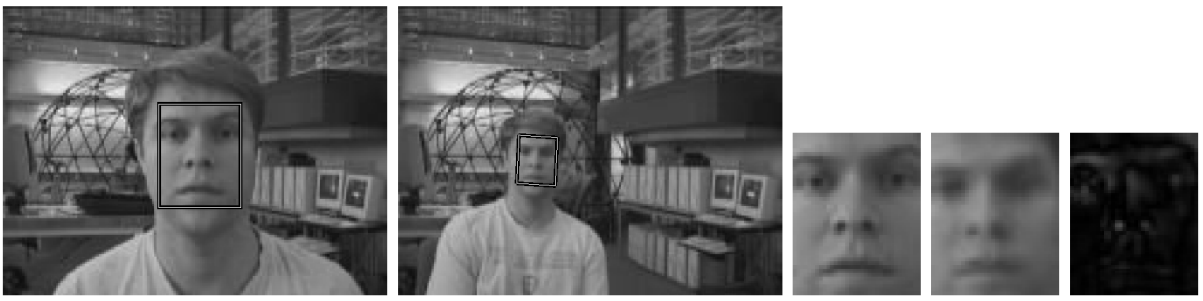


Fig. 2. Textured region. (a) Selecting the reference image for a face tracker. (b) One possible state. (c) Reference image $\mathbf{I}_R$ from (a). (d) Normalized comparison image $\mathbf{I}_C$ for the state in (b). (e) Difference image $|\mathbf{I}_R - \mathbf{I}_C|$.

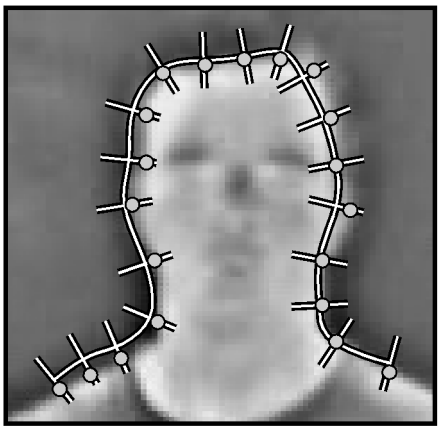


Fig. 3. Snake. State of a sample snake head tracker (infrared image). Circles on curve normals indicate locations of strongest Sobel edges.

A snake is represented as a periodic or nonperiodic cubic B-spline [17], [21], [22] constrained to deform affinely. The spline approach allows an arbitrarily detailed description of the shape of the tracked object, while the affine constraint efficiently captures the snake's degrees of freedom if its associated contour is a rigid, planar curve restricted to translation, scaling, and in-plane rotation. The image prediction function π for snakes hypothesizes a curve derived from the current affine parameters $\mathbf{Q}$ along which there is an intensity disparity. To compute the image similarity between the image and this prediction, we define $p(\mathbf{I} \mid \mathbf{X})$ by adapting the formula for "$p(z \mid x)$," as described in [23].

For each of the n segment borders comprising the B-spline parametrized by a particular $\mathbf{Q}$, edge detection is performed along a line of length L (typically 10-20 pixels) that is normal to and bisected by the curve at that point. Let $\mathbf{\Lambda}(i)$ be the image location of the curve at segment i, where $0 \leq i < n$. Using the Canny algorithm, we let $\mathbf{z}(i)$ be the location of the edge segment along the ith normal that is found nearest to $\mathbf{\Lambda}(i)$. For the Sobel method, $\mathbf{z}(i)$ is the strongest edge along the normal whose strength is over the threshold τ. The shape of a nonperiodic snake and the Sobel edges detected on its normals are illustrated in Fig. 3.

Assuming the state $\mathbf{X}$ includes $\mathbf{Q}$, we express the likelihood as:

$$p_{snake}(\mathbf{I} \mid \mathbf{X}) = \exp\left(-\frac{1}{\sigma^2_{snake}} \sum_{i=0}^{n-1} l(i) \cdot \psi_{snake}(i)\right), \quad (6)$$

where $l(i)$ is the fraction of the total length $|\mathbf{\Lambda}|$ of the snake represented by normal i. The degree to which the location of each detected edge fits the shape model is given by

$$\psi_{snake}(i) = \begin{cases} |\mathbf{\Lambda}(i) - \mathbf{z}(i)| & \text{if an edge is found} \\ \xi & \text{otherwise,} \end{cases} \quad (7)$$

ξ serves as a penalty value for $\psi(i)$ when there is no edge detected along the ith normal.

4 Tracking a Single Object

In this section, we discuss techniques for tracking single, unoccluded objects that are *atomic* in the sense that they are identified by only one of the modalities presented in the previous section. The combination of an identifying modality and the observable parameters of that modality (size, color, shape, etc.) constitute an *attribute* of an object. To visually track an atomic object, we want to follow the area of the image that is the best match to it. A filter such as the Kalman filter [3], [4] can be used to predict the most likely location and other characteristics of this area, indicating where to begin searching for it. In the first part of this section, we discuss methods for finding and parametrizing a set of hypotheses for good matches. The best match thus found is suitable for input to a standard Kalman filter as the measurement. Later in the section, we examine the Probabilistic Data Association Filter (PDAF) [3], an extension to the Kalman filter that considers other highly likely alternatives.

4.1 The Measurement Process

The measurement extraction process is essentially a search for maxima of the image likelihood $p(\mathbf{I} \mid \mathbf{X})$ in the neighborhood of $\widehat{\mathbf{X}}$, the state predicted from the filter at time t. The geometric characteristics (such as $x \in X, y \in Y$, orientation $\phi \in \Phi$, and scale $s \in S$) of the image areas corresponding to these maximally likely states are derived as measurements $\mathbf{Z}$. Perhaps the simplest class of suitable techniques are gradient ascent methods such as conjugate gradient and Powell's method [2]. Vision-specific forms of gradient ascent are often used to efficiently obtain a single best measurement with which to update a tracking filter: For example, with snakes [18], [17] and textured regions [16], [24], [25]. Gradient ascent works best, however, when $p(\mathbf{I} \mid \mathbf{X})$ is unimodal. The object state must also be changing slowly enough that the filter can keep up if the algorithm is terminated after a maximum number of steps (i.e., it only gets a fraction of the way to the true maximum for each new image), or if the posterior is multimodal that the predicted state will not wind up in another basin of attraction. Another assumption if there is multimodality is that there will not be significant interference between modes, such as two modes (the correct one and an incorrect one) merging and splitting, creating the possibility of an incorrect choice after the split.

These difficulties with gradient methods are why in many situations we favor other algorithms that allow for faster state changes and multiple modalities in the state posterior. Randomized methods such as the factored sampling approach of the Condensation algorithm [23] have proven successful at finding nonlocal maxima of multimodal image likelihoods. Accordingly, we use a measurement generation method which we call *measurement sampling* which is adapted from factored sampling but retains the notion of locality around a single predicted state. Intuitively, we sample points in state space from the prior distribution on the state $p(\mathbf{X})$, compute their image likelihoods $p(\mathbf{I} \mid \mathbf{X})$, throw away all but the top fraction, and derive the measurement parameters of what remains. N samples are taken from a normal distribution in the target's state space $\mathcal{X}$ centered on its current predicted state $\widehat{\mathbf{X}}$. N and the covariance of the distribution $\Sigma_{\mathcal{X}}$ are chosen to give adequate coverage to a "tracking window" about the target. $p(\mathbf{I} \mid \mathbf{X})$ is computed for each sample by scoring the

(a) (b) (c)

Fig. 4. Measurement sampling for textured regions. (a) Samples with small covariance. (b) Top fraction of samples becomes measurements. (c) Best measurement after thinning by gradient ascent and enforcement of minimum separation (see Section 5.1.1).

degree of fit between the hypothesized target and the current image. Finally, a winnowing step sorts the samples by their likelihoods and keeps only the n most likely ones $(n \ll N)$ for input to the tracking filter.

The measurement sampling process for a textured region tracker constrained to translation is shown in Fig. 4. The object of interest is the face of a baseball player in a team photograph. For simple translation of the region, state space is $\mathcal{X} = X \times Y$. In Fig. 4a, $N = 250$ states are sampled about the predicted face location with a sampling covariance of $\Sigma_{\mathcal{X}} = \mathrm{diag}\,(100, 100)$ and the $n = 5$ best are selected as measurements in Fig. 4b. Fig. 4c is explained in the next section.

4.2 Probabilistic Data Association Filter

Kalman filtering [3], [4] is an efficient method for tracking when the distribution on measurements is Gaussian (see the Appendix for a brief overview). Situations in which there are departures from the assumption that the posterior is Gaussian, however, require extensions to the Kalman filter. For example, noise might temporarily create multiple measurements or cause the target-originated measurement to disappear. Or, we might be tracking T objects as independent entities and, thus, expect there to be a persistent measurement for each one. Proper target-measurement correspondences are maintained by continually computing the *association probabilities* of the various possibilities.

The measurement processes described above derive a group of candidate states for each tracker. The Probabilistic Data Association Filter (PDAF) [3], [26] is an extension of the Kalman filter [3] that uses a Bayesian approach to the problem of data association or how to update the state when there is a single target and possibly no measurements or multiple measurements due to noise. Rather than possibly erring by choosing the *nearest neighbor* [3], [26], or data closest to what is expected in order to update the state, the PDAF hedges its bets by weighting the influence of the various candidate measurements based on two assumptions. First, it assumes that there is exactly one target giving rise to one "true" measurement which may sporadically disappear either because the target is temporarily occluded or because of suboptimal feature detection at any stage of the pipeline between the camera and (for example) the edge detection algorithm. Second, the PDAF assumes that all other measurements are "false" and arise from a uniform noise process.

The relevant step in the Kalman filter is the computation of the innovation $\boldsymbol{\nu}$. The PDAF introduces a notion of the *combined* innovation, computed over the n measurements detected at a given time step as the weighted sum of the individual innovations: $\boldsymbol{\nu} = \sum_{i=1}^{n} \beta_i \boldsymbol{\nu}_i$. Each β_i is the probability of the *association event* Θ_i that the $i\mathrm{th}$ measurement is target-originated. Also computed is β_0, the probability of the event that none of the measurements is target-originated (i.e., the target is associated with the null measurement). These events encompass all possible interpretations of the data, so $\sum_{i=0}^{n} \beta_i = 1$.

Sampling randomly from a normal distribution and selecting the top fraction of the samples as measurements does not precisely satisfy the PDAF assumption of a uniform distribution of false measurements, but it is usually a reasonable approximation. Multiple measurements coming from the true, target-originated peak in the image likelihood function $p(\mathbf{I} \mid \mathbf{X})$ tend to be tightly clustered in one part of state space. Because the effect of the PDAF association probabilities is to average the contribution of the measurements to the state estimate, measurements closely arranged around a maximum of $p(\mathbf{I} \mid \mathbf{X})$ harmlessly average out to that maximum.

5 Tracking Multiple Objects Jointly

The PDAF tracker in the previous section assumes that there are no other strong, persistent features in the image that have attributes similar to those of the tracked target. False peaks in $p(\mathbf{I} \mid \mathbf{X})$ can be due to actual noise sources—e.g., capture hardware or unpredictable phenomena like rustling leaves or highlights on a rippling water surface. However, many scene elements—other *parts* of a compound object being tracked, a static background, other moving objects, etc.—may engender strong enough peaks that measurements from them will be generated disproportionately, biasing the PDAF filter's state estimates. If, for example, the states of the multiple parts become proximate, one target-originated measurement may be claimed by another target. Simply running a separate PDAF tracker on each part could lead to multiple trackers locked onto the same part.

This section introduces new methods for dealing with this class of problems by tracking all of the image features that may mutually cause distraction and adding a layer of logic to ensure that trackers are correctly distributed over the measurements. One such technique that we discuss is an existing extension to the PDAF called the Joint Probabilistic Data Association Filter (JPDAF) [3]. The first part of the section investigates the issues involved in adapting the JPDAF to vision; one limitation is that it can only be used for groups of objects of the same modality. In the second half of the section, we introduce a new approach called the Joint Likelihood Filter (JLF). The JLF captures the crux of the JPDAF but is applicable to mixtures of tracking modalities, is more efficient than the JPDAF, and reasons about occlusion relationships between objects.

5.1 Joint Probabilistic Data Association Filter

The Joint Probabilistic Data Association Filter (JPDAF) [3], [26] enforces a kind of exclusion principle that prevents two or more trackers from latching onto the same target by calculating target-measurement association probabilities jointly. Suppose that we are tracking T objects, for which a total of n measurements have been generated from the current image (methods for deriving measurements for all objects *jointly* are presented below). A key notion in the JPDAF is that of a *joint event* Θ or conjunction of association events Θ_{jt_j} (the subscript t_j denotes which target measurement j is matched to). The probability of a particular Θ depends, as with the PDAF, on the distances between each target's predicted measurement and the actual measurement it is associated with in Θ. However, an additional influence on the probability of Θ stems from the interaction of the various association events in Θ. If the measurement process generates, at most, one measurement for each peak in the image likelihood function $p(\mathbf{I} \mid \mathbf{X})$ and each target induces at most one peak in $p(\mathbf{I} \mid \mathbf{X})$, two kinds of combinations of associations are logically *infeasible*. First, a joint event Θ containing two associations $\Theta_{jt_1}, \Theta_{jt_2}$ such that $t_1 \neq t_2$ and $j \neq 0$, implies that two different targets are responsible for the same measurement, a contradiction. Second, if Θ includes associations $\Theta_{it_i}, \Theta_{jt_j}$ such that $i \neq j$ but $t_i = t_j$, this amounts to an interpretation that a single target has spawned multiple measurements—also an impossibility. The JPDAF disregards infeasible joint events and, thus, avoids inappropriate state convergence. The precise formula for the probability of each particular target-measurement association is given in [3], [26].

5.1.1 JPDAF Measurement Generation

A desirable characteristic of joint measurement generation is that only one measurement be created for each peak in $p(\mathbf{I} \mid \mathbf{X})$. Random sampling alone typically extracts multiple measurements not attributable to noise for each target, which violates one of the presumptions of the JPDAF and can lead to multiple targets becoming incorrectly associated with that peak. We address these issues by introducing a single, joint measurement process over all targets that apportions measurements rationally.

The joint method we use for T targets is based on the random sampling technique presented in the previous section. After eliminating low-fitness samples per the process previously described, each remaining sample $\mathbf{Z}_i$ is "improved" using conjugate gradient ascent [2] to obtain a local maximum $\mathbf{Z}'_i$. The purpose of the hill-climbing step is twofold. First, the resulting samples $\mathbf{Z}'_i$ are more consistent, reducing error in the state estimate. Second, states that are on the slopes of the same peak of $p(\mathbf{I} \mid \mathbf{X})$ but separated by the randomness of the sampling process tend to converge in state space $\mathcal{X}$ as they ascend (provided certain local conditions on $p(\mathbf{I} \mid \mathbf{X})$ hold). Thus, we deduce that aggregations of samples after hill-climbing will be relatively tightly clustered around local maxima, allowing the selection of the best sample in each cluster as representative of a peak.

The last step is therefore to try to choose one exemplar for each group of samples. This is done by enforcing a *minimum separation* between samples in $\mathcal{X}$. Starting with the most fit sample $\mathbf{X}_{best}$, all less fit samples $\mathbf{X}_i$ such that $|\mathbf{X}_{best} - \mathbf{X}_i| \leq \Delta$ are eliminated. In practice, we use a different threshold Δ_k for each parameter of the joint measurement and eliminate samples which are too close along any dimension. (Unless otherwise noted, we use $\Delta_X = \Delta_Y = 10$ pixels, $\Delta_\Phi = 0.1$ radians, and $\Delta_S = 0.01$). The purpose of Δ is to compensate for any lack of precision in the hill-climbing algorithm. The thinning process is repeated for the next fittest sample and so on, yielding a set of n measurements generally equal to the number of tracked objects T. The value of n can vary due to the randomness of the sampling procedure and whether the image actually has only T target-like features.

This method is applied in Fig. 4c to the baseball team picture from the previous section, resulting in a single measurement.

5.2 Joint Likelihood Filter

The JPDAF, though a useful advance over the PDAF, lacks certain desirable properties. First, due to its requirement that every tracker have the same image likelihood $p(\mathbf{I} \mid \mathbf{X})$ (so that any candidate image feature for one tracker can be plausibly associated with any other), the JPDAF is inapplicable to mixtures of different kinds of trackers. Second, the measurement generation process outlined can encounter difficulties when targets overlap one another. This is because of the JPDAF's assumption that the image likelihoods of multiple objects are independent when they actually are not. Consider the analog of (1) for multiple object states (assuming conditioning on previous images):

$$p(\mathbf{X}_1, \ldots, \mathbf{X}_T \mid \mathbf{I}) = kp(\mathbf{I} \mid \mathbf{X}_1, \ldots, \mathbf{X}_T)p(\mathbf{X}_1, \ldots, \mathbf{X}_T).$$

The last term on the right hand side, which we call the *joint state prior*, is embodied in the JPDAF by the joint feasibility logic in its formula for association probabilities [3]. However, thus far, we have assumed that the first term on the right hand side, which we call the *joint image likelihood*, can be factored as

$$p(\mathbf{I} \mid \mathbf{X}_1, \ldots, \mathbf{X}_T) = p(\mathbf{I} \mid \mathbf{X}_1) \cdots p(\mathbf{I} \mid \mathbf{X}_T).$$

Evaluating image likelihoods independently is an approximation that tends to break down when targets are very close or overlapping because this is exactly when their appearances become dependent on one another. When object A occludes or abuts object B, it affects expectations about the appearance of object B and at least part of the immediate background of both objects. Ignoring this effect can introduce a systematic bias in the position, angle, or scale estimate that leads to mistracking. To track objects more accurately, $p(\mathbf{I} \mid \mathbf{X}_1, \ldots, \mathbf{X}_T)$ must consider the ordering of the depths, relative to the camera, of the tracked objects. Knowing which object is in front of which when they overlap is the key to properly predicting the image's appearance $\pi(\mathbf{X}_1, \ldots, \mathbf{X}_T)$ from the objects jointly.

The joint image likelihood effectively functions as the joint event probability of the JPDAF since it encodes a measurement association (as well as the likelihood of that measurement) for every target. However, by sampling the prior in state space for each tracker we can build up a joint measurement $\mathbf{Z}^J$ and directly assess its likelihood without incurring the combinatorial penalty associated with the JPDAF. Repeating this joint sampling step yields a pool of joint samples. We call the process that results from these changes the Joint Likelihood Filter (JLF). Details are presented in the next two sections.

5.2.1 Joint Measurement Process

The first step in the *joint measurement* process of the JLF is to generate N joint samples. A given joint sample $\mathbf{X}_i^J$, $1 \leq i \leq N$, is built from T *component samples* $\mathbf{X}_j$, $1 \leq j \leq T$, each generated by one of the trackers in its state space $\mathcal{X}_j$. The component sampling process is the same as that used by PDAF and JPDAF trackers: A sample is generated either randomly from the distribution defined by the predicted state $\widehat{\mathbf{X}}_j$ and sampling covariance $\Sigma_{\mathcal{X}_j}$, or nonrandomly (when, for example, pure gradient ascent is being used). The component samples are then stacked to get a joint sample: $\mathbf{X}_i^J = (\mathbf{X}_1, \ldots, \mathbf{X}_T)^T$, so $\mathcal{X}^J = \mathcal{X}_1 \times \ldots \times \mathcal{X}_T$. Associations, in the JPDAF sense, are implicit: target j is associated with component sample $\mathbf{Z}_j$.

The second step for each joint sample is to pick the most likely depth ordering of its T component samples. To do this, all permutations of depth orderings are enumerated, tagging each component sample with a depth order index in the process. Different depth orderings of nonoverlapping component samples are visually equivalent, inducing equivalence classes of depth orderings, so we automatically eliminate all but one representative of each class. Let $\mathbf{D}_{\mathbf{X}_i^J} = \{\mathbf{d}_1, \ldots, \mathbf{d}_{K_{\mathbf{X}^J}}\}$ be the set of visually distinct depth order permutations of joint sample $\mathbf{X}_i^J$. For efficiency, we only do gradient ascent on the most likely depth ordering $\mathbf{d}_i$ of each joint sample (a *joint image likelihood* objective function is described in the next section) rather than all of them. Finally, the most probable of all of the joint samples $\mathbf{X}_i^J$ is selected and converted to a *joint measurement* $\mathbf{Z}^J$. The component measurements $\mathbf{Z}_1, \ldots, \mathbf{Z}_T$ of $\mathbf{Z}^J$ are then plugged into Kalman filters for their associated trackers.

An example of a joint sample comprising a textured region and a snake is shown in Fig. 5a. The textured region is tracking a chess pawn and the snake is tracking a knight. Since there are two overlapping component samples in the joint sample of the chess example referred to above, there are two depth ordering hypotheses. Hypotheses corresponding to the pawn being in front of the knight and the knight being in front of the pawn are represented in Figs. 5b and 5e, respectively.

5.2.2 Joint Image Likelihood

To evaluate the likelihood of a particular joint sample $\mathbf{X}_J$ and its depth ordering $\mathbf{D}_{\mathbf{X}^J}$, the probabilities of its component samples are computed *jointly*. A key difference between this operation and the independent approach of the PDAF and JPDAF is our ability to predict occlusions between objects. When one object is hypothesized to be in front of another, expectations about the occluded object's appearance change. Trackers of snakes will not expect edges where they are blocked from view, homogeneous region trackers will not expect occluded pixels to fit the color model, and so on. Specifically, $\mathbf{D}_{\mathbf{X}^J}$ allows us to *mask* [27], [28] occluded portions of objects such that the occluding objects take precedence in the formation of a jointly predicted image $\pi(\mathbf{X}_1, \ldots, \mathbf{X}_T)$. Pixels predicted to be obstructed are ignored and those predicted to be visible are matched normally.

A basic technique of the independent image likelihoods in Section 3 is to compute a mean match value ψ over the extent or around the perimeter of the object. Under the JLF, the set of masks $\{\mathbf{M}_j\}$ is used to modify this technique for two reasons. First, some pixels are erroneously counted more than once by the PDAF and JPDAF when tracked objects overlap; each pixel should only be used as evidence by one tracker. Second, the masks are used to try to ensure that each pixel is counted by the correct tracker. An approach that meets these criteria only counts target pixels that are predicted to be visible in the calculation of that target's mean match value. The masking procedure induced by $\mathbf{D}_{\mathbf{X}^J}$ outputs a binary mask $\mathbf{M}_j$ the size of the image $\mathbf{I}$ for each target t_j. $\mathbf{M}_j(x, y) = 1$ indicates that the image pixel $\mathbf{I}(x, y)$ comes from target t_j and $\mathbf{M}_j(x, y) = 0$ indicates that the pixel belongs to either another object or the background. $\mathbf{M}_{knight}$ is shown for the two depth ordering hypotheses of the chess example in Figs. 5c and 5f. $\mathbf{M}_{pawn}$ is shown for those two hypotheses in Figs. 5d and 5g. Note the alteration in shape of the mask when an object is partially occluded.

For a textured region t_j, only those interior pixels (x, y) for which $\mathbf{M}_j(x, y) = 1$ contribute to the mean match value. That is, portions of the region's interior that are not visible do not have a match value computed and are subtracted from the effective area. This method is illustrated for the textured-region pawn of the chess example in Figs. 5h, 5i, and 5j. Fig. 5h shows the reference image for the pawn. Figs. 5i and 5j show the comparison images for the hypotheses that the pawn is in front of and behind the knight, respectively. In the latter case, the nearer knight masks out the area of pixels shown in black. Homogeneous regions are slightly more subtle. The central area is handled in the same fashion as textured regions, but the inhibitory frame is not in the mask $\mathbf{M}_j$ of the tracker. Rather, only those pixels (x, y) in the inhibitory frame for which

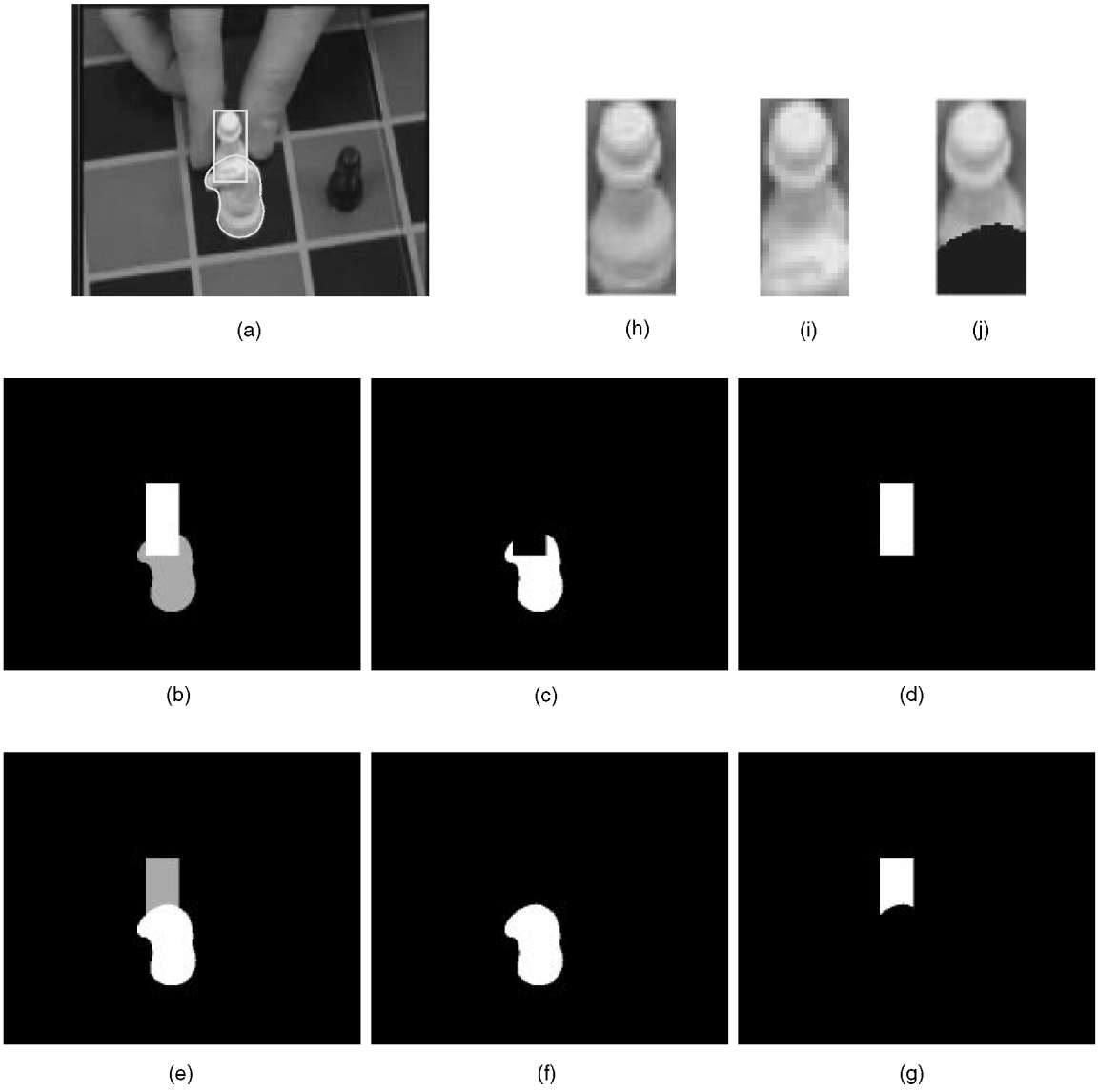


Fig. 5. Joint likelihood filter: Depth orderings. (a) Joint measurement, (b) first depth ordering, (c) first knight mask, (d) first pawn mask, (e) second depth ordering, (f) second knight mask, (g) second pawn mask, (h) pawn reference image, (i) first pawn comparison image, and (j) second pawn comparison image.

$\mathbf{M}_i(x,y) = 0$ for all $i \neq j$ are counted. The same method is also used for snakes: Only edges found at locations (x,y) such that $\mathbf{M}_i(x,y) = 0$ for all $i \neq j$ are considered. Finally, any pixels in the interior, frame, or on the normals of an object that are also outside of the image are treated as masked out.

It is also important to guard against interpreting an object as being completely occluded when there is image evidence for its visibility. This problem can be avoided by classifying visible pixels as either positive or negative evidence for the hypothesis that the target is in a certain state and putting masked pixels in a third, neutral category rather than ignoring them. What makes a pixel a *match* or positive evidence instead of negative, is fundamentally a threshold Υ in ψ. To quantify this approach, matching pixels are assigned a value of 1, nonmatching pixels a value of -1 and masked pixels get 0. Measurements with more corroborative evidence are assigned higher likelihoods than those with no or negative evidence by using the sigmoid function on the sum of the pixel match values.

Specifically, we replace the independent image likelihoods $p(\mathbf{I} \mid \mathbf{X})$ for homogeneous regions, textured regions, and snakes from Section 3 with component image likelihoods $p^J(\mathbf{I} \mid \mathbf{X}_j)$. For textured regions, we have:

$$p^J_{tregion}(\mathbf{I} \mid \mathbf{X}_j) = \text{sig}\left(\sum_{\text{x,y}\in\mathbf{I}_\text{R}} a(x,y) \cdot \psi^J_{tregion}(x,y)\right), \tag{8}$$

where

$$\psi^J_{tregion}(x,y) = \begin{cases} 1 & \text{if } \mathbf{M}_j(x,y) = 1 \wedge (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2 \leq \Upsilon_{tregion} \\ -1 & \text{if } \mathbf{M}_j(x,y) = 1 \wedge (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2 > \Upsilon_{tregion} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The form of the component image likelihood is analogous for homogeneous regions and snakes; we omit them due to space limitations. More details can be found in [11].

Let the joint tracker, which has T component trackers, consist of a set $\mathcal{H}$ of homogeneous region trackers, a set $\mathcal{T}$ of textured region trackers, and a set $\mathcal{S}$ of snake trackers such that $T = |\mathcal{H}| + |\mathcal{T}| + |\mathcal{S}|$. With the component image likelihoods defined as above, the image likelihood of the joint sample $\mathbf{X}^J$ is simply their product:

$$p^J(\mathbf{I} \mid \mathbf{X}^J) = \prod_{t_j \in \mathcal{H}} p^J_{hregion}(\mathbf{I} \mid \mathbf{X}_j) \prod_{t_j \in \mathcal{T}} p^J_{tregion}(\mathbf{I} \mid \mathbf{X}_j) \prod_{t_j \in \mathcal{S}} p^J_{snake}(\mathbf{I} \mid \mathbf{X}_j). \quad (10)$$

It is straightforward to perform gradient ascent on the joint image likelihood to improve the component samples. Note that gradient ascent does not change the depth ordering of the component samples, however.

6 Tracking Linked Objects with Constraints

An important assumption of the preceding algorithms is that occlusions and distractions are caused by other tracked objects or visual phenomena reasonably approximated by noise. When this expectation is violated, as when such occurrences are actually due to persistent features of the visual environment, these tracking filters can yield biased results or mistrack. In this case, the selection of what area of the target to focus on and what tracking modality to use becomes paramount in determining tracking accuracy.

With regard to making this selection, it is useful to distinguish between an object *attribute*, as defined in Section 4, and what we call a *part*. A part is a spatially distinct image feature physically linked to the larger object. Fundamentally, a part is *what* a tracker tracks, while an attribute is *how* the tracker identifies its target. We have observed that the more an object is occluded or the better a distracting background feature matches an attribute used for tracking, the more severe the deterioration of accuracy and the greater the chance of outright failure of a PDAF/JPDAF tracker. The approach of this section to the problem of persistent distractors is to try to reduce their incidence and, hence, their influence, by defining a target as a conjunction of parts and/or attributes. An atomic tracker with temporarily weak discriminatory power can overcome difficult image conditions because of the *constraints* imposed by its linkage to other trackers. These force consideration of the entire ensemble of parts and attributes simultaneously when interpreting the image, helping to rule out incorrect alternatives. Constraints are only applicable, of course, when we are tracking a target complex enough that it has multiple resolvable parts and/or attributes.

A linkage between targets means that they are parts of some larger object and that their states are therefore not independent. This disallows the decomposition of the joint state prior $p(\mathbf{X}_1, \ldots, \mathbf{X}_T) = p(\mathbf{X}_1) \cdots p(\mathbf{X}_T)$ that is a vital step in both the JPDAF and JLF multiple-object tracking algorithms. As with the joint image likelihood $p^J(\mathbf{I} \mid \mathbf{X}^J)$ in the previous section, we need a more complex formulation of $p(\mathbf{X}_1, \ldots, \mathbf{X}_T)$ that takes into account the interactions between objects by describing how multiple *linked* objects influence one another's states, even at a distance.

In the next part of this section, we introduce an extension to the JLF called the Constrained Joint Likelihood Filter, or CJLF, that implements interpart constraints efficiently and simply. We then present results demonstrating how the CJLF improves tracking performance in many visual situations over the previously described algorithms and enables certain tracking tasks to be carried out for which those algorithms are not suited.

6.1 Constrained Joint Likelihood Filter

The expectation that parts or attributes of a complex tracked object will be in a particular configuration is extra information that may help distinguish the object from the background or other objects. The key idea behind the CJLF is an elaboration of one of the most basic kinds of constraints: limitation of the number of parameters in an object's state, which in turn, reduces the size of its measurement space. We already use this form of constraint for atomic trackers when we analyze the object, the tracking task and the visual environment in order to decide what geometric parameters to estimate. If the object to be tracked only slides back and forth horizontally, for example, or rotates in place, then there is no reason to give the tracker more than the minimal degrees of freedom required to follow that class of movement. To do otherwise only provides the tracker with an opportunity to mistrack along an extraneous state dimension.

For a multipart or multiattribute object, there are multiple trackers for which this kind of decision must be made. The CJLF simply formalizes the common sense notion that a minimal state description of the entire object implies certain correlations between and limitations on the states of its constituent parts and attributes. Ordinarily, a special-purpose tracker with a customized image likelihood function $p(\mathbf{I} \mid \mathbf{X})$ is created for tracking a complicated object. The CJLF avoids this by providing a small set of rules for composing atomic trackers such that the joint image likelihood is a product of component likelihoods. The rationale for this decision is twofold: 1) to reduce the amount of time spent on analysis and code writing for novel tracking tasks by permitting code reuse and 2) to provide a standard interface for new methods to easily be integrated with existing ones.

The compositional primitives used by the CJLF are based on intuitive physical relationships such as rigid links, hinges, and fixed depth orderings. Given a set of parts or attributes with unconstrained state spaces $\mathcal{X}_1, \ldots, \mathcal{X}_T$, these rules serve as a guide for paring them down to their minimal, constrained forms: $\mathcal{X}'_1, \ldots, \mathcal{X}'_T$. When the paring removes all degrees of freedom of a tracker, its state space becomes empty. It is still desirable to perform image processing for that tracker, so as a matter of bookkeeping, the notion of the tracker is retained. This process is the primary method by which constraints are introduced into the joint state prior. In addition to reducing the degrees of freedom available to some of the trackers, the CJLF's compositional rules also indicate how to derive the image processing variables of linked parts from one another. The

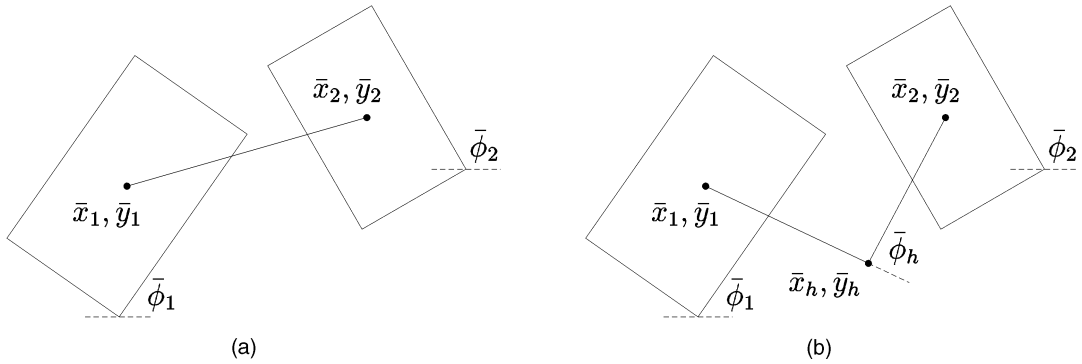


Fig. 6. Constraint types. (a) Initial configuration of a rigid link. (b) Initial configuration of a hinge.

details of this derivation are explicated for each of the rules in the next section.

For purposes of implementation, the CJLF approach alters the method of obtaining geometric image processing parameters from the state referred to in Section 4. Let each target t_j have a *measurement key* $\mathbf{K}_j$. Previously, the domain of each function in $\mathbf{K}_j$ was implicitly$\mathcal{X}_j$; we now extend it to the joint state space $\mathcal{X}^J$. This allows us to refer to the component measurement geometric parameters of *any* target t_i to define t_j's component measurement geometric parameters. The effect of this reduction in the joint state space is to alter the JLF so that it considers *only* those joint state samples which satisfy the constraints exactly, allowing their joint probabilities to be computed normally. Sampling and hill-climbing can then be used as in the previous section, while still meeting the conditions on the interrelationship of the parts.

6.1.1 Constraint Types

Rigid link constraints. The simplest kind of constraint between measurements is a *rigid link*. A rigid link between two objects t_1, t_2 implies that t_2's current geometric parameters are completely determined by their initial values and t_1's current values—it has no state or measurement space of its own. Its only function is to contribute to the calculation of the joint image likelihood $p(\mathbf{I} \mid \mathbf{X}_1, \mathbf{X}_2)$. Therefore, t_2 does not use a Kalman filter to estimate its own state; its purpose is as an adjunct that makes t_1 a more complex visual object. As an example, suppose that two rigidly linked objects are allowed to translate, scale, and rotate, and that the initial offset between them scales as they do. This joint object configuration is diagrammed in Fig. 6a. The details of the mathematics are trivial but tedious, so we omit them here [11]. As shorthand, we represent the rigid link transformation that takes the geometric parameters of object i to those of object j as a function $R_{i,j}$. Thus, $\mathbf{K}_2 = R_{1,2}(\mathbf{K}_1)$.

It is straightforward to generalize a two-part, rigidly constrained joint object to a T-target system. T rigidly linked parts can be modeled by treating them as $T-1$ linked pairs, every one of which includes target t_1, such that $\mathbf{K}_i = R_{1,i}(\mathbf{K}_1)$.

Hinge constraints. A more complex constraint is a *hinge*, which is like a rigid link but with an angular degree of freedom granted to the second object; the axis of rotation is determined by the initial image location of the hinge: $\bar{x}_h, \bar{y}_h$ (see the diagram in Fig. 6b). The equations of the two-part joint object from above, allowing the ensemble to translate, scale, and rotate freely and the second part to rotate independently about the hinge, are also covered in [11]. The hinge transformation between objects i and j is denoted by $H_{i,j}$.

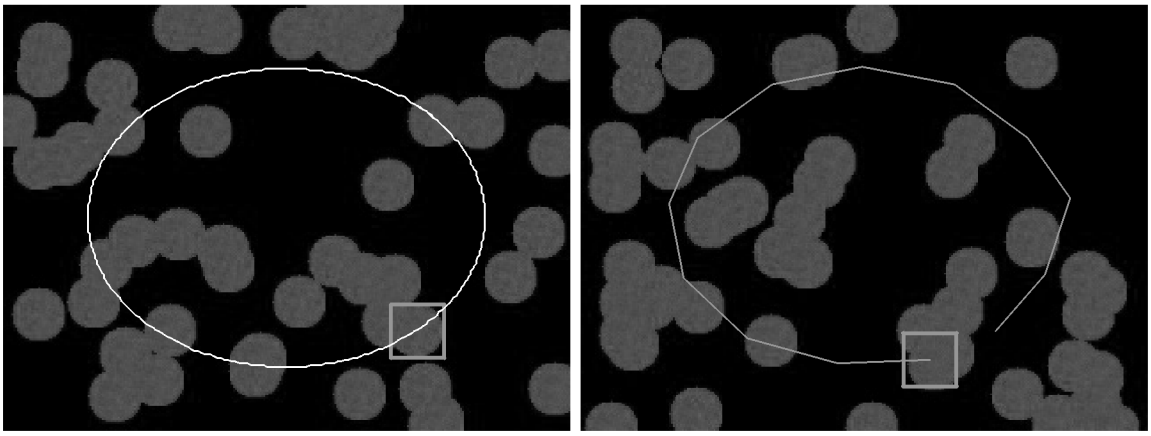
We can also extend the mathematics of a single hinge constraint to a system of multiple hinges. T parts connected in sequence by $T-1$ hinges form a *chain* [29]. Let C be a chain consisting of T hinge-connected parts: $C = (t_1, \ldots, t_T)$. We can specify the constraint on each part along C inductively: If the first and second links t_1, t_2 are defined by the two-part system introduced above, then the state of the ith part for $i > 1$ is $\mathbf{X}_i = (\phi_i)$ and its measurement space is $\mathcal{Z}_i = \Phi$. Given the measurement key $\mathbf{K}_1$ of the first part t_1, the measurement key of the ith part t_i is given by

$$\mathbf{K}_i = H_{i-1,i}(H_{i-2,i-1}(\ldots H_{1,2}(\mathbf{K}_1)\ldots)).$$

By writing $H_{i-1,i}(\mathbf{K}_{i-1})$, the calculations that lead to $\mathbf{K}_{i-1}$ are assumed.

Depth constraints. Another useful kind of constraint is related to depth. When there is an expectation that some subset of the objects being tracked will not occlude one another, we can collect them into a *depth group*. Objects in the same depth group are not masked against one another during computation of the joint image likelihood. When justified, grouping objects in this way is more efficient because there are fewer depth orderings to consider for each joint measurement.

An obvious situation to which depth groups apply occurs when tracking an object with multiple attributes. Since attributes represent qualities of a physical object rather than the object itself, multiple instances can be "layered" onto a single object without affecting the visibility of any of them. When a person's face, for example, is tracked by both a textured region tracker (to capture appearance) and a homogeneous region tracker (for skin color), the two trackers are members of the same depth group. Depth groups are also appropriate for parts linked by constraints under certain viewing and motion conditions. Though these parts are spatially distinct, if they are physically prevented from overlapping, they can also be placed in the same depth group.



(a) (b)

Fig. 7. PDAF: Tracking a homogeneous region with uniform noise (CG). (a) Frame 0 with initial position and ground truth of path overlaid. (b) Frame 300 with history of estimates at 25 frame intervals.

7 Results

In the following sections, we present some results for our probabilistic tracking algorithms. For purposes of exact comparison between techniques, all input sequences are MPEGs unless otherwise indicated.

7.1 Tracking Objects Independently

In Fig. 7, the use of multiple measurements by the PDAF to increase the noise resistance of the tracker is illustrated. A CG sequence was created in which a single red circle moves counterclockwise at a rate of 0.02 radians per frame, while 50 distracting red circles are uniformly randomly placed in each frame. The target state is $\mathbf{X} = (x, y)$ and the measurement parameters are the same; it is tracked with a homogeneous region tracker. $N = 100$ samples are chosen with a sampling covariance of $\Sigma_{\mathcal{X}} = \text{diag}\,(100, 100)$. In one series of experiments, only the best sample ($n = 1$) was used as a measurement: The tracker was able to follow the circle through a full orbit in five out of 20 trials. In another series of experiments, $n = 10$ measurements were selected. This tracker was much less vulnerable to distraction and succeeded in tracking the circle in 17 out of 20 trials.

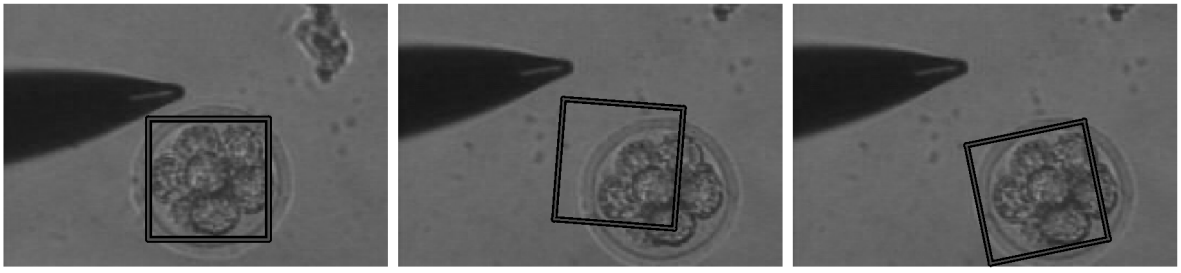
Fig. 8 shows how using random sampling for measurement generation can yield more robust performance than pure gradient ascent when there are agile motions. A textured region tracker is attached to a mouse embryo as the microscope slide is moved and the embryo is poked with a probe. The state of the tracker is position and orientation: $\mathbf{X} = (x, y, \phi)$, and measurement space is $\mathcal{Z} = X \times Y \times \Phi$. A tracker that uses gradient ascent (Powell's method) alone to generate a single measurement is thrown off when the embryo moves abruptly after frame 60. A tracker that uses random sampling for measurement generation, however, recovers from these agile motions $(N = 250, n = 5, \Sigma_{\mathcal{X}} = \text{diag}\,(100, 100, 0.04))$.

Fig. 9 shows a homogeneous region tracker following the forearm of a person as he walks from left to right. The state includes the forearm's image position, orientation, and the velocities of these parameters: $\mathbf{X} = (x, y, \phi, \dot{x}, \dot{y}, \dot{\phi},)$. Each measurement is a translation and rotation of a fixed size rectangle, so $\mathcal{Z} = X \times Y \times \Phi$. The rectangles overlaid on the figure indicate the measurements $(N = 1,000, n = 10, \Sigma_{\mathcal{X}} = \text{diag}\,(100, 100, 0.02))$.

In Fig. 10, we track two human heads in infrared (IR) imagery as they primarily translate and scale, one with a closed contour and the other with an open curve. Thus, the state of each tracker is expressed as $\mathbf{X} = (x, y, s)$ and $\mathcal{Z} = X \times Y \times S$ $(N = 250, n = 5, \Sigma_{\mathcal{X}} = \text{diag}\,(100, 100, 0.01))$.

7.2 Tracking Objects Jointly

Fig. 11 demonstrates the efficacy of the JPDAF vs. the PDAF for tracking the faces of two people as they cross paths.



(a) (b) (c)

Fig. 8. PDAF measurement generation: Tracking a mouse embryo with a translating, rotating textured region. (a) Intitial state. (b) Gradient ascent mistracks by frame 120 due to excessive speed. (c) Random sampling is successful through from 120. (Sequence courtesy of G. Danuser).

Fig. 9. PDAF: Tracking a swining arm with a translating, rotating homogeneous region. Measurements $(n = 10)$ generated by random sampling are shown. (Sequence courtesy of J. MacCormick).

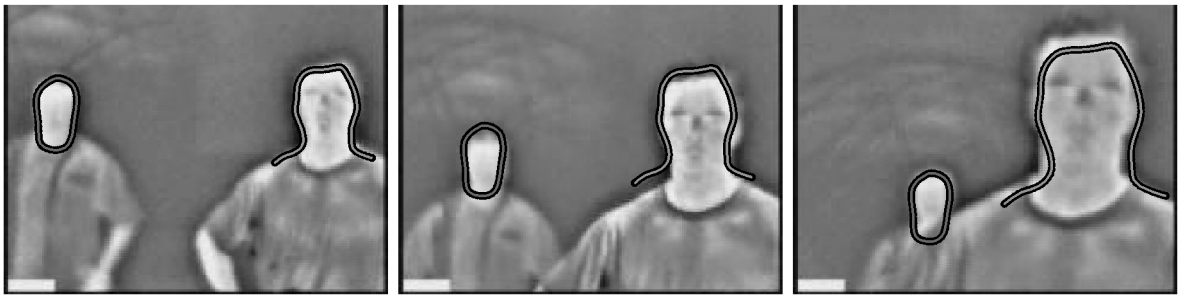


Fig. 10. PDAF: Tracking two faces with translating, scaling snakes (Infrared imagery, Canny).

Using translating homogeneous regions with identical dimensions and the same skin color model, the state of each tracker is $\mathbf{X} = (x, y, \dot{x}, \dot{y})$, making measurement space $\mathcal{Z} = X \times Y$. Both the PDAF and the JPDAF tracker select the best 10 of 50 samples, where the state sampling covariance is $\Sigma_{\mathcal{X}} = \text{diag}\,(100, 100)$. Each remaining sample is improved with conjugate gradient ascent, and a minimum separation is enforced —independently for the PDAF but jointly for the JPDAF. The JPDAF successfully tracked both heads through the crossing in 10 out of 10 trials, whereas the PDAF failed in 10 out of 10 trials. In every case, the tracker assigned to the head of the person walking to the left was distracted by the rightward-moving head.

The ability of the JLF to infer the depth ordering of tracked objects is illustrated in Fig. 12. A white pawn chess piece is tracked by a textured region as it briefly moves behind a white knight, which is tracked by a snake; both have state $\mathbf{X} = (x, y, \dot{x}, \dot{y})$, making each component's measurement space $\mathcal{Z} = X \times Y$. Measurement generation is done using pure gradient ascent with Powell's method. The tracker's outline, normally white, is drawn in gray when the most likely depth ordering indicates that it is partially occluded. The fact that the pawn is behind the knight during the middle section of the tracking sequence is correctly deduced.

7.3 Tracking Objects with Constraints

Now suppose we want to track the pawn from the previous example without knowing about the knight. Using a JLF tracker consisting solely of a homogeneous region initialized as shown in frame 0 of Fig. 13a[1], the state is $\mathbf{X} = (x, y)$, making measurement space $\mathcal{Z} = X \times Y$. The single most likely of 50 samples from a state sampling covariance of $\Sigma_{\mathcal{X}} = \text{diag}\,(50, 50)$ is selected and improved with Powell's method. This approach fails because the untracked white knight fits the color model well and attracts the pawn strongly. The fundamental problem is the presence of a strong, persistent peak due to the knight in the homogeneous region's image likelihood that is not expected by the JLF tracker.

Tracking the pawn in a similar fashion with a snake alone yields better results because $p_{snake}(\mathbf{I} \mid \mathbf{X})$ has only one prominent extremum rather than two. This quantifies our intuition that shape is a better cue for this task than color. Without knowing ahead of time which modality, if any, is sufficiently distinctive for successful tracking, a prudent strategy is to use multiple attributes simultaneously. The conjunction of color and shape results in a joint image likelihood $p^J(\mathbf{I} \mid \mathbf{X}^J)$ with peaks only where *both* likelihoods $p_{hregion}(\mathbf{I} \mid \mathbf{X})$ and $p_{snake}(\mathbf{I} \mid \mathbf{X})$ have peaks, reducing distractions. Formally, we utilize the pawn's color and shape simultaneously by modeling it with two rigidly linked attributes: A homogeneous region and a snake with coincident centers. The pawn's joint region-snake tracker follows the same regime of hill-climbing on the single best of 50 samples as the single-attribute trackers above. As

1. A single-object JLF is not the same as a standard PDAF tracker because of the way match values are computed in the joint image likelihood $p^J(\mathbf{I} \mid \mathbf{X}^J)$, but we use the JLF here to make comparisons with the CJLF clearer.

Fig. 11. JPDAF vs. PDAF: Tracking crossing homogeneous regions. Frames 0, 12, and 24 are shown.

Fig. 13b shows, this constrained formulation permits the pawn to be successfully tracked when the homogeneous region alone fails.

Another example of tracking with the CJLF is given in Fig. 14. In the input sequence, a person walks from the left side of the frame slightly toward the camera and then in profile to the right. Suppose we want to track the person's face as a homogeneous region with a single-part JLF tracker. The state is $\mathbf{X} = (x, y, \dot{x}, \dot{y}, s)$ and measurement space is $\mathcal{Z} = X \times Y \times S$; the best single sample of 50 is improved using Powell's method, where $\Sigma_{\mathcal{X}} = \mathrm{diag}\,(50, 50, 0.001)$. Because of a somewhat skin-colored brick wall in the background, the discriminatory power of the face tracker is marginal. The face tracker is distracted by a column of tan bricks in the center of the image; when the person emerges on the other side of the bricks in frame 120 of Fig. 14a, tracking has failed. This occurs for essentially the same reason as with the pawn tracking example above: The bricks are unmodeled, very similar to the target, and in close proximity to it for too many frames.

A tracker with the same filter parameters can track the red shirt through the same sequence without any problems, however, because its color is much more distinctive the face's color. Exploiting the physical connection of the face to the shirt, we track the two as rigidly linked parts that scale and translate together. Fig. 14b shows a successful run. The CJLF tracker sometimes bobbles slightly in front of the brick column as the tracker explores the possibility of not translating anymore and instead simply expanding to including the bricks, the face, and the shirt. Because of the nonlocality of random sampling, however, this part of measurement space is quickly discarded as the proportion of nonmatches in the larger area dilutes its fitness compared to the correct interpretation.

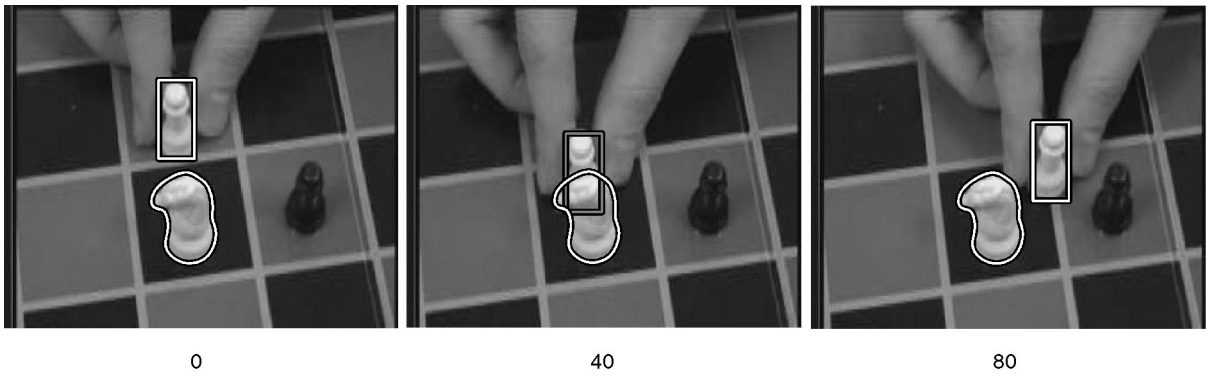
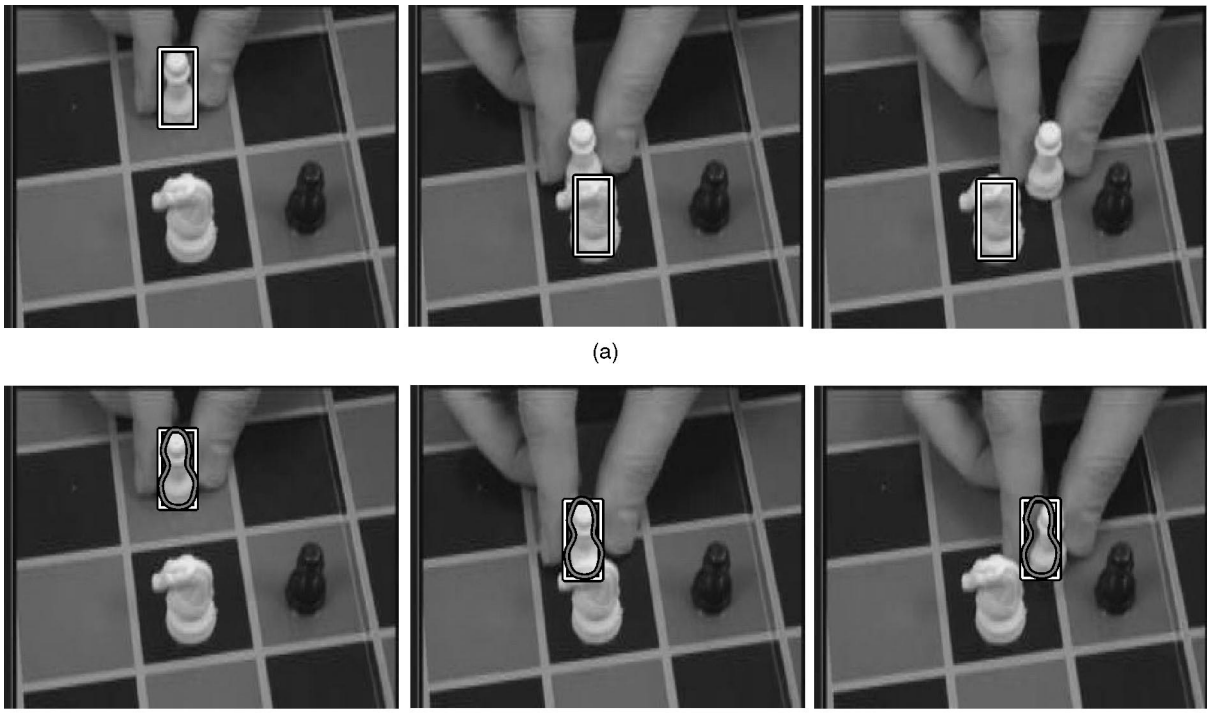


Fig. 12. JLF: Deducing the occlusion relationship between a textured region and snake.

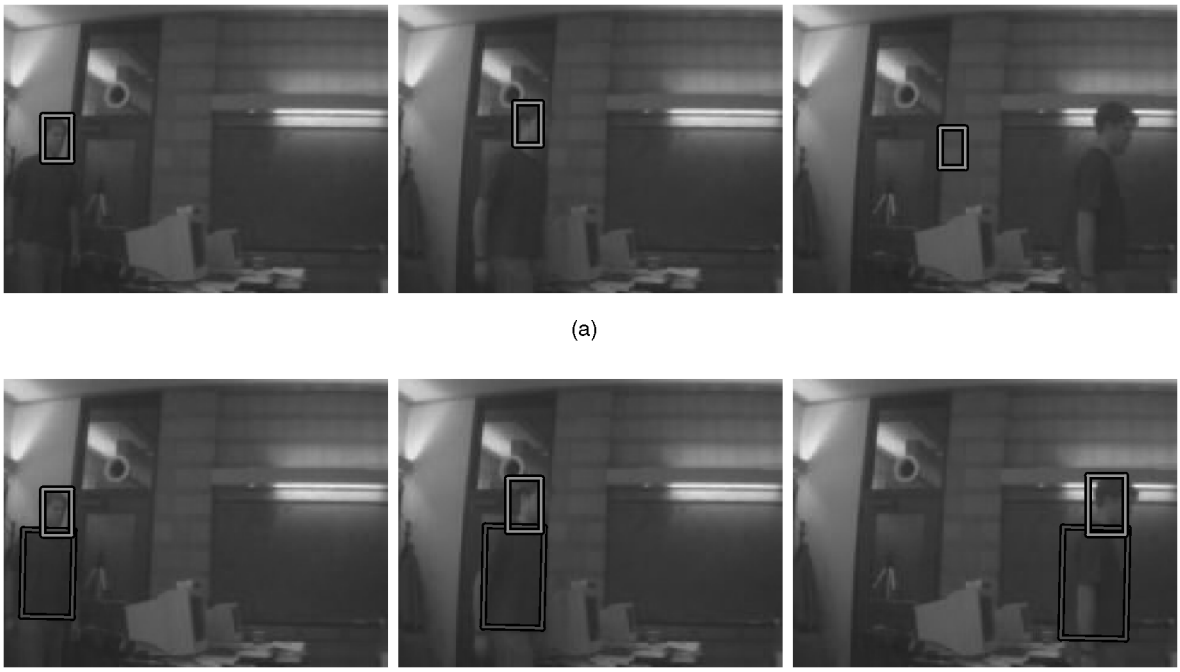


(a)

(b)

Fig. 13. Multiattribute CJLF (frames 0, 50, and 100). (a) One-attribute JFL tracker. JLF homogeneous region tracker is distracted by the white knight. (b) Two-attribute CJLF tracker. CJLF homogeneous region and snake tracker overcomes the distraction.

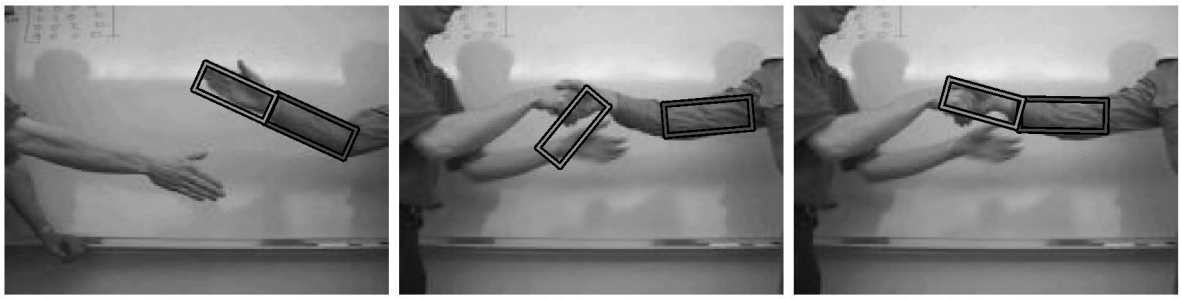
A more complicated situation which shows the advantage of the CJLF over the JLF is shown in Fig. 15. Here, we want to track a person's hand and forearm as homogeneous regions while they shake hands with another person, who is not tracked. Each component ($i = 1, 2$) of the JLF tracker has a state of the form $\mathbf{X}_i = (x_i, y_i, \phi_i, \dot{x}_i, \dot{y}_i, \dot{\phi}_i)$ with measurement spaces $\mathcal{Z}_1 = \mathcal{Z}_2 = X \times Y \times \Phi$. Accelerations during the handshake are too large for pure gradient tracking, so each



(a)

(b)

Fig. 14. Multipart CJLF: Resisting a distracting background (frames 0, 60, and 120). (a) One-part JLF tracker on the face is distracted. (b) Two-part CJLF tracker on the face and shirt succeeds.



(a) (b) (c)

Fig. 15. CJLF: Using a hinge constraint at the wrist to prevent mistracking during a handshake. (a) Frame 0 of sequence of homogeneous region trackers on hand and forearm. (b) Running a JLF tracker for both parts, the hand tracker is distracted by other person's hand (frame 260). (c) The CJLF formulation permits accurate tracking of the hand (frame 260).

component tracker selects the best 1 of 50 samples, where $\Sigma_{\mathcal{X}} = \mathrm{diag}\,(50, 50, 0.002)$ and hill-climbs it using Powell's method. Despite these measures, the hand tracker mistracks when its target is in close proximity to the other person's hands, and the forearm tracker erroneously slides along the sleeve.

These shortcomings can be eliminated with a hinge constraint joining the hand and forearm trackers to one another at the midpoints of their abutting short sides. The state of the forearm tracker remains the same, while the hand tracker is reduced to one degree of angular freedom. Adopting this approach prevents the hand and forearm trackers from floating apart; relatively higher joint image likelihoods keep the hinge at the sleeve-hand border. The result is that during the period of ambiguity when the two hands are clasped together, a realistic interpretation of the situation is maintained and tracking proceeds correctly after the hands separate.

8 Related Work

Most of the previous work on tracking complex objects has not explicitly tackled the data association issue. One line of primarily motion-based tracking work has avoided the association or correspondence problem entirely through a differential approach. For example, Yamamoto and Koshikawa [30], tracked in-plane articulated movements of a human arm by relating arm motion to image change via the Jacobian and solving the brightness equation using least-squares. Basu et al. [31] used a similar technique to recover 3D head motion parameters, and other assemblages of body parts have been tracked in [28], [32], [33], [29]. Many of these efforts have more of a flavor of pure estimation, rather than the simultaneous problem of estimation and label assignment that we focus on.

The Condensation algorithm [23] tackles the problem of clutter by maintaining a set of hypotheses about associations that are resolved over time. It also uses random sampling, but lacks an explicit notion of state. Rather, the samples must be queried to obtain one. The query procedure provided does not work well when the image likelihood is multimodal and the authors suggest that a more sophisticated "mode finder" is necessary. This is essentially what our measurement generation algorithm of Section 5.1.1 implements.

The difficulties arising from mutual occlusions among tracked objects have been addressed by a number of heuristic extensions to the Kalman filter. For example, Rehg [28] tracked the fingers of a human hand as they bent and blocked one another; and Koller et al. [27] tracked the outlines of cars on a highway as they sometimes occluded one another. The essential idea of both approaches was to mask out the occluded part to prevent it from claiming the measurement generated by the occluding part. In each case, 3D information was available to predict which part was occluded, whereas the Joint Likelihood Filter we introduced deduces occlusion relationships from the image directly.

	PDAF	JPDAF	JLF	CJLF
Generate samples	N from $(\hat{\mathbf{X}}; \mathbf{\Sigma}_{\mathcal{X}})$	N_j from $(\hat{\mathbf{X}}_j; \mathbf{\Sigma}_{\mathcal{X}_j})$ $\forall\ T$ targets	N from $(\hat{\mathbf{X}}^J; \mathbf{\Sigma}_{\mathcal{X}^J})$	N from $(\hat{\mathbf{X}}'_1, \ldots, \hat{\mathbf{X}}'_T;$ $\mathbf{\Sigma}_{\mathcal{X}'_1}, \ldots, \mathbf{\Sigma}_{\mathcal{X}'_T})$
Evaluate	$p(\mathbf{I}\|\mathbf{X})$ $\forall$ samples	$p(\mathbf{I}\|\mathbf{X})$ $\forall$ samples of each target	$p^J(\mathbf{I}\|\mathbf{X}^J)$ over $\mathbf{D}^J$ $\forall$ joint samples	$p^J(\mathbf{I}\|\mathbf{X}^J)$ over $\mathbf{D}^J$ $\forall$ joint samples
Hill-climb	Optional	Yes	Optional	Optional
Winnow	n best samples	Best samples separated by $\mathbf{\Delta}$	Best joint sample	Best joint sample

Fig. 16. Tracking algorithm steps. N is the number of samples, n is the number of measurements, T is the number of targets. $(\mu; \sigma)$ indicates a Gaussian with mean μ and covariance σ. $\mathbf{X}'$ and $\mathcal{X}'$ are minimal, constrained forms of a state and state space, respectively.

The JLF is similar to work described in [34]; a data association approach to tracking is also taken in [35].

9 Conclusion

This paper's primary contribution is its demonstration of the importance of reasoning about correspondences between trackers and image data in order to achieve robust vision-based tracking. Though filters such as the PDAF and JPDAF were originally developed for discrete radar and sonar tracking applications, we were able to successfully adapt them to visual tasks by defining measurements suitably and devising a novel preprocessing step to extract them. Run head-to-head on the same image sequences, the vision-based tracking algorithms thus created exhibited markedly better performance in the presence of clutter and when tracking multiple identical objects than many current commonly-used methods.

We have also explicated shortcomings in the JPDAF and remedied them with a more efficient and sophisticated method, the JLF. By relating the exclusion principle at the heart of the JPDAF to the method of masking out image data, the JLF handles occlusions between tracked objects. Our extension of this method to collections of objects of different modalities such as color, shape, and appearance is original. The approach we take to color representation and region geometry for homogeneous regions is our own. Moreover, though others have used three-dimensional state parameters to assist with occlusion reasoning, the JLF's inference of the depth ordering of tracked objects from image data alone is novel.

Finally, we augmented the JLF method to allow low-level trackers to be composed via part and attribute constraints in order to specify more complex targets. This algorithm, the CJLF, reduces the vulnerability of a vision-based tracker to unmodeled distractions and occlusions by effectively defining its target more distinctively. Although geometric constraints are a well-established method for increasing robustness, exploiting multiple modalities simultaneously to track a single object—especially three, as we do—is fairly new, and the union of these two approaches is clearly an advance. The way that the CJLF framework does so is made more useful by its flexibility and extensibility: Target models can be easily specified and new modalities can be added straightforwardly.

In future work, we hope to rationalize the selection of visual cues used for object tracking based on image conditions, and to allow for persistent distractors to be found automatically and tracked as objects in their own right instead of being treated as noise.

Appendix

The Kalman filter [3] estimates a time-varying state $\mathbf{X}$ from observable measurements $\mathbf{Z}$ of a system which at time t is described by the dynamic equation $\mathbf{X}_t = \mathbf{F}\mathbf{X}_{t-1} + \mathbf{q}_t$, where $\mathbf{q}_t$ is a sequence of zero-mean, white, Gaussian noise with dynamic covariance $\mathbf{Q}$. The state is related to $\mathbf{Z}$ by the measurement equation $\mathbf{Z}_t = \mathbf{H}\mathbf{X}_t + \mathbf{r}_t$, where $\mathbf{r}_t$ is also Gaussian noise with measurement covariance $\mathbf{R}$. Using the previous state estimate and the current data, a new estimate for $\mathbf{X}$ is generated as follows (except for the identity matrix $\mathbf{I}$, every variable not subscripted by $t-1$ is implicitly subscripted by t):

$\hat{\mathbf{X}}=\mathbf{F}\mathbf{X}_{t-1}$	Predicted state	$\hat{\mathbf{Z}}=\mathbf{H}\hat{\mathbf{X}}$	Predicted measurement
$\hat{\mathbf{P}}=\mathbf{F}\mathbf{P}_{t-1}\mathbf{F}'+\mathbf{Q}$	State prediction covariance	$\mathbf{S}=\mathbf{H}\hat{\mathbf{P}}\mathbf{H}'+\mathbf{R}$	Measurement prediction covariance
$\boldsymbol{\nu}=\mathbf{Z}-\hat{\mathbf{Z}}$	Innovation	$\mathbf{W}=\hat{\mathbf{P}}\mathbf{H}'\mathbf{S}^{-1}$	Filter gain
$\mathbf{X}=\hat{\mathbf{X}}+\mathbf{W}\boldsymbol{\nu}$	State estimate	$\mathbf{P}=(\mathbf{I}-\mathbf{W}\mathbf{H})\hat{\mathbf{P}}$	State covariance estimate.

The first-order Extended Kalman Filter [3] handles the case of a nonlinear dynamic equation $\mathbf{X}_t = F(\mathbf{X}_{t-1}) + \mathbf{q}_t$ through linearization by assigning the first term of the Taylor series expansion of F about $\mathbf{X}_{t-1}$ at each filter update to the matrix $\mathbf{F}$. A nonlinear measurement equation $\mathbf{Z}_t = H(\mathbf{X}_t) + \mathbf{r}_t$ is dealt with similarly by expanding H about $\hat{\mathbf{X}}$ every filter update to obtain $\mathbf{H}$.

Acknowledgments

This research was done while C. Rasmussen was at the Department of Computer Science, Yale University.

References

[1] J. Mendel, *Lessons in Digital Estimation Theory.* Prentice-Hall, 1987.
[2] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge Univ. Press, 1993.
[3] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.
[4] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.,* vol. 82, pp. 35-45, 1960.
[5] D. Mumford, "Pattern Theory: A Unifying Perspective," *Perception as Bayesian Inference,* D. Knill and W. Richards, eds., pp. 25-62, Cambridge Univ. Press, 1996.
[6] B. Ripley, *Pattern Recognition and Neural Networks.* Cambridge Univ. Press, 1996.
[7] M. Isard and A. Blake, "Condensation-Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, pp. 5-28, 1998.
[8] D. Knill, D. Kersten, and A. Yuille, "Introduction: A Bayesian Formulation of Visual Perception," *Perception as Bayesian Inference,* D. Knill and W. Richards, eds., pp. 1-21, Cambridge Univ. Press, 1996.
[9] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics-Principles and Practice.* Addison-Wesley, 1989.
[10] C. Rasmussen and G. Hager, "Tracking Objects by Color Alone," Technical Report DCS-RR-1114, Yale Univ., 1996.
[11] C. Rasmussen, "Integrating Multiple Visual Cues for Robust Tracking," PhD thesis, Yale Univ., New Haven, Conn., 2000.
[12] G. Klinker, S. Shafer, and T. Kanade, "A Physical Approach to Color Image Understanding," *Int'l J. Computer Vision,* vol. 4, pp. 7-38, 1990.
[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.
[14] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, 1981.
[15] J. Shi and C. Tomasi, "Good Features to Track," *Proc. Conf. Computer Vision and Pattern Recognition,* 1994.
[16] G. Hager and P. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 10, pp. 1025-1039, Oct. 1998.
[17] A. Blake, M. Isard, and D. Reynard, "Learning to Track the Visual Motion of Contours," *Artificial Intelligence,* no. 78, pp. 101-133, 1995.
[18] D. Terzopoulos and R. Szeliski, "Tracking with Kalman Snakes," *Active Vision,* A. Blake and A. Yuille, eds., pp. 3-20, MIT Press, 1992.

[19] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, 1986.
[20] I. Sobel, "An Isotropic 3×3 Image Gradient Operator," *Machine Vision for Three-Dimensional Scenes,* H. Freeman, ed., pp. 376-379, Academic Press, 1990.
[21] J. Hoschek and D. Lasser, *Fundamentals of Computer-Aided Geometric Design.* A.K. Peters, 1993.
[22] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques.* Addison-Wesley, 1992.
[23] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. European Conf. Computer Vision,* pp. 343-356, 1996.
[24] B. Horn, *Robot Vision.* MIT Press, 1986.
[25] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. European Conf. Computer Vision,* pp. 237-252, 1992.
[26] I. Cox, "A Review of Statistical Data Association Techniques for Motion Correspondence," *Int'l J. Computer Vision,* vol. 10, no. 1, pp. 53-65, 1993.
[27] D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 189-196, 1994.
[28] J. Rehg and T. Kanade, "Model-Based Tracking of Self-Occluding Articulated Objects," *Proc. Int'l Conf. Computer Vision,* pp. 612-617, 1995.
[29] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," *Proc. Computer Vision and Pattern Recognition,* pp. 8-15, 1998.
[30] M. Yamamoto and K. Koshikawa, "Human Motion Analysis Based on a Robot Arm Model," *Proc. Conf. Computer Vision and Pattern Recognition,* 1991.
[31] S. Basu, I. Essa, and A. Pentland, "Motion Regularization for Model-Based Head Tracking," *Proc. Int'l Conf. Pattern Recognition,* 1996.
[32] D. Morris and J. Rehg, "Singularity Analysis for Articulated Object Tracking," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 289-296, 1998.
[33] A. Pentland and B. Horowitz, "Recovery of Nonrigid Motion and Structure," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 7, pp. 730-742, July 1991.
[34] J. MacCormick and A. Blake, "A Probabilistic Exclusion Principle for Tracking Multiple Objects," *Proc. Int'l Conf. Computer Vision,* 1999.
[35] T. Cham and J. Rehg, "A Multiple Hypothesis Approach to Figure Tracking," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 239-245, 1999.

Christopher Rasmussen received the AB degree in computer science from Harvard College in 1993. He received the MS and PhD degrees in computer science from Yale University in 1995 and 2000, respectively, where he studied techniques for real-time visual tracking with Dr. Gregory D. Hager. Currently, he is a National Research Council Postdoctoral Research Associate in the Perception Systems Group at the National Institute of Standards and Technology, Gaithersburg, Maryland. His research interests include vision-based tracking, autonomous robot navigation, and sensor fusion.

Gregory D. Hager received the BA degree, summa cum laude, in computer science and mathematics from Luther College in 1983, and the MS and PhD degrees in computer science from the University of Pennsylvania in 1985 and 1988, respectively. From 1988 to 1990, he was a Fulbright junior research fellow at the University of Karlsruhe and the Fraunhofer Institute IITB in Karlsruhe, Germany. From 1991 until 1999, he was with the Computer Science Department at Yale University. In 1999, he joined the Computer Science Department at Johns Hopkins University where he is now a full professor and a faculty member in the Center for Computer Integrated Surgical Systems and Technology. Professor Hager has authored more than 100 research articles and books in the area of robotics and computer vision. His current research interests include visual tracking, vision-based control, medical robotics, and human-computer interaction. He is a member of the IEEE and the Computer Society.