A Unifying Theoretical Framework for Region Tracking

Kevin Cannons and Richard P. Wildes

Technical Report CSE-2013-04

February 8 2013

Department of Computer Science and Engineering
4700 Keele Street, Toronto, Ontario M3J 1P3 Canada

A Unifying Theoretical Framework for Region Tracking

Kevin Cannons
Richard P. Wildes

Department of Computer Science and Engineering
and the Centre for Vision Research
York University
Toronto, Ontario M3J 1P3
Canada

February 8, 2013

Abstract

Visual region-based tracking is a heavily researched general approach to following a target across a temporal image sequence. Little research, however, has addressed the interrelationships of the various proposed approaches at a theoretical level. In response to this situation, the present paper describes a unifying framework for a wide range of region trackers in terms of the amount of spatial layout that they maintain in their target representation. This framework yields a general notation from which any of these trackers can be instantiated. To illustrate the practical utility of the framework, a range of region trackers are instantiated within its formalism and used to document empirically the impact of maintaining variable amounts of spatial information during target tracking.

Contents

Chapter 1

Introduction

Object tracking is one of the most heavily researched areas in all of computer vision. Within this literature, region trackers currently receive particularly notable attention owing to their wide applicability and strong performance in empirical evaluation [1]. In essence, region trackers are distinguished by their use of dense, area-based representations that characterize the target support and thereby can be contrasted with other classes of trackers that rely on sparser representations (e.g., discrete feature and contour trackers [1]). Interestingly, even though region trackers have been the subject of intensive investigation, little has emerged in terms of overall frameworks that *theoretically relate* the various region tracking approaches. The current paper takes strides to fill this gap in the literature by presenting a unifying framework for a wide range of region trackers. Such a framework can serve to enhance understanding of commonalities and differences between extant approaches as well as provide a mechanism for developing and analyzing new, state-of-the-art systems.

Conceptually, much of the region tracking literature can be reviewed by considering the degree to which the various approaches maintain spatial organization of their primitive visual measurements (e.g., colour, texture and motion) across target support, see Fig. 1.1, where several representative examples are highlighted. At one extreme, basic kernel histogram trackers operate by maintaining a single distribution of measurements that is aggregated across the entire target region (e.g., [2, 3]), thus sacrificing all spatial layout information. Benefits of utilizing such a coarse representation include potential speed improvements due to the reduced dimensionality of the representation as well as increased flexibility (e.g., for tracking during non-rigid

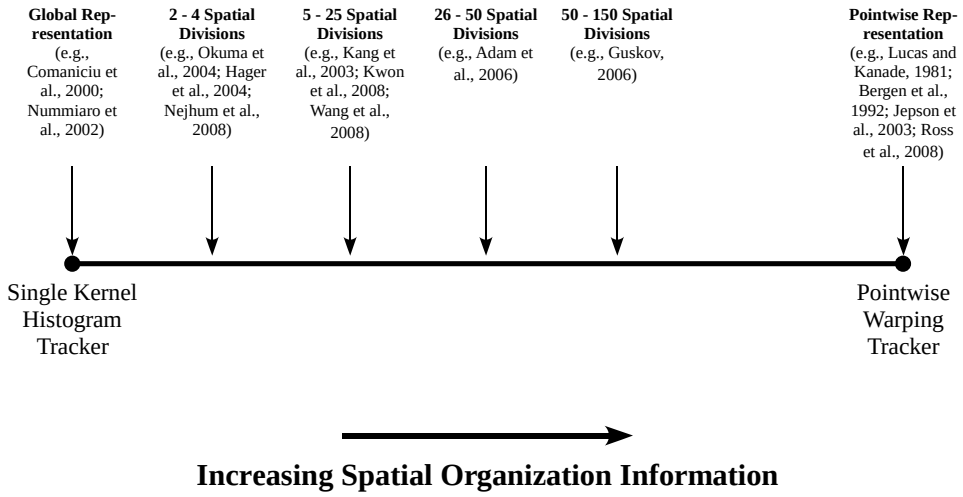

Figure 1.1: A continuum of region trackers that vary based on the amount of spatial organization information that is Retained throughout the tracking process. Examples of various trackers are listed along the continuum.

deformations). Other systems incorporate very limited spatial arrangement information (e.g., [4, 5, 6, 7, 8, 9, 10]). For example, Okuma et al. propose a system for tracking hockey players where different color distributions are maintained for the top and bottom halves of the players' bodies [4]. Still other approaches subdivide the target even more finely (e.g., [11, 12, 13, 14, 15]). With an increased number of sample locations along with smaller accumulation regions, these systems retain additional spatial organization information regarding the target. *FragTrack* is a well known example of such a system that represents the target using roughly 36 histograms derived from overlapping target spatial regions [14]. Finally, at the opposite end of this continuum that considers the amount of retained target spatial organization lie systems traditionally termed template warp trackers (e.g., [16, 17, 18, 19, 20, 21]). These trackers retain complete information regarding the spatial layout of the target and the maximum number of data points with which to perform effective matching by employing pointwise feature representations. The region of accumulation in such systems is minimal, subtending just a single pixel. Perhaps most closely related to the current paper is a joint feature-spatial spaces tracker that models uncertainty in spatial layout by varying the bandwidth of a *single* spatial kernel. Also related is an approach that models spatial layout transformations during tracking via a noise model [22].

The current contribution differentiates itself by explicitly modeling variation in the number, placement, and size/bandwidth of tracking kernels as well as presenting a systematic evaluation of these parameters on several real-world videos.

Given the potential for organizing a significant portion of the region tracking literature in terms of the amount of spatial organization that is maintained across target support, the remainder of this paper adds to the literature in two important ways. (i) It provides a novel analytic framework that formally unifies a diversity of region trackers according to the amount of maintained target spatial organization information. This framework connects a range of *existing* trackers as opposed to presenting another single approach to visual tracking in isolation. (ii) It empirically explores this space of trackers defined by the framework to document scenarios where it is beneficial to operate with different amounts of spatial layout information. This ability to systematically compare region trackers based on the amount of retained spatial information is made practically possible due to the first contribution, the unifying framework.

Chapter 2

Analytic framework

To show how this continuum of trackers, ranging from kernel histogram to template warp systems, can be unified under a common theoretical framework, two specific trackers will be derived that define its end points. These derivations will be performed with the goal of making the two tracker update equations that serve to map a maintained target template to a target candidate in a particular image within a video as similar as possible. Consideration will be restricted to image data represented in the form of *multichannel* feature measurements (e.g., RGB color or outputs of spatial filters at multiple orientations [19]), as a generalization of single channel measurements (e.g., raw image intensity). Initially, particular choices will be made regarding key tracker components, including the error function and optimization procedure. Sum-of-squared differences (SSD) will be used as the error function to be optimized in a gradient fashion, e.g., [23]. Subsequently, these specific choices will be relaxed to yield a more general framework. The following notational conventions are adopted. Variables that appear within the multiple kernel histogram derivation will appear with an overline (e.g., $\bar{\mathbf{x}}_{\bar{n}}$); analogous variables within the template warping tracker framework will appear with no overlines (e.g., $\mathbf{x}_n$). Variables that are common to both tracking architectures will have no overlines (e.g., time, t).

Multichannel multiple kernel histogram tracker. In the kernel histogram approach to target tracking, one or more spatial kernels, K, are used to weight the relative importance of pixels across the target support (e.g., [2, 5]). Multiple kernels within a target's representation will be indexed by the variable $\bar{n}$ and centered about point $\bar{\mathbf{x}}_{\bar{n}}$. Also employed is a simple function, b, that maps a pixel's intensity to a histogram bin, m. Then, a

multiple kernel histogram tracker defines the target template, consisting of a set of histograms, as

$$\hat{q}_m\left(\bar{\mathbf{x}}_{\bar{n}}\right) = C_q \sum_{j=1}^{J} K\left(\mathbf{x}_j - \bar{\mathbf{x}}_{\bar{n}}\right) \delta\left(b\left(\mathbf{x}_j\right) - m\right), \tag{2.1}$$

which is compared to target candidates defined at time t, specified as

$$\hat{p}_m\left(\bar{\mathbf{x}}_{\bar{n}}, \mathbf{a}, t\right) = C_p \sum_{j=1}^{J} K\left(\mathbf{x}_j - \left(\bar{\mathbf{x}}_{\bar{n}} + \mathsf{W}\left(\bar{\mathbf{x}}_{\bar{n}}\right)\mathbf{a}\right)\right) \delta\left(b\left(\mathbf{x}_j\right) - m\right), \tag{2.2}$$

where C_q and C_p are normalization terms to ensure the histograms sum to unity, as is common practise when processing histogram data (e.g., [2], [14]). Additionally, δ is the Dirac delta function, and j varies to consider all J pixels within the kernel support. To capture the spatial mapping between a candidate in an image frame and the target template, $\hat{p}_m$ includes a warping function, W, with parameters, $\mathbf{a}$, that models the allowed geometric transformation (motion) of the target between images (e.g., translation, affine, etc.).

In the above histogram definitions, kernel weighting is performed upon single channel data; whereas, here the goal is to consider multichannel features. When operating on multichanneled data, a *modified histogram* construction technique is used whereby each bin in the histogram corresponds to a particular channel of the data [24]. Each pointwise feature vector increments a histogram bin based on the magnitude of the corresponding feature channel. Histograms constructed in the latter fashion illustrate the relative presence or absence of each feature channel over the target support. This modified template histogram is defined as

$$\hat{q}_m\left(\bar{\mathbf{x}}_{\bar{n}}\right) = C_q \sum_{j=1}^{J} \left(K\left(\mathbf{x}_j - \bar{\mathbf{x}}_{\bar{n}}\right) Q\left(\mathbf{x}_j, m\right)\right), \tag{2.3}$$

where the normalization term is defined according to

$$C_q = \frac{1}{\sum_{m_i=1}^{M} \sum_{j=1}^{J} \left(K\left(\mathbf{x}_j - \bar{\mathbf{x}}_{\bar{n}}\right) Q\left(\mathbf{x}_j, m_i\right)\right)}, \tag{2.4}$$

$Q\left(\mathbf{x}_j, m_i\right)$ is the multichannel feature data in the first frame of the tracking sequence, and m_i varies over all $M > 1$ channels of feature data. Additionally, the features are indexed according to their particular spatial location, $\mathbf{x}_j$.

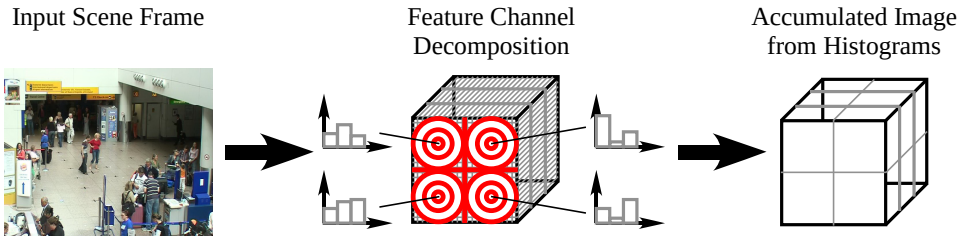

Figure 2.1: Creation of an "accumulated image" with a multiple kernel histogram tracker. (left) Input image from a video sequence. (middle) Feature accumulation using four kernels denoted by red circles. The feature representation spans three channels; thus, a three-binned histogram is constructed for each kernel. (right) The four histograms, with their spatial relationships preserved via reference to an image grid, can be viewed as an accumulated image with reduced resolution.

The updated target candidate histogram is defined as

$$\hat{p}_m\left(\bar{\mathbf{x}}_{\bar{n}}, \mathbf{a}, t\right) = C_p \sum_{j=1}^{J} \left(K\left(\mathbf{x}_j - \left(\bar{\mathbf{x}}_{\bar{n}} + \mathsf{W}\left(\bar{\mathbf{x}}_{\bar{n}}\right)\mathbf{a}\right)\right) P\left(\mathbf{x}_j, m, t\right)\right), \tag{2.5}$$

where $P\left(\mathbf{x}_j, m, t\right)$ is the multichannel feature data at frame t and the normalization term, C_p, is defined analogously to C_q, (2.4), ensuring $\hat{p}_m$ sums to unity.

Conceptually, when manipulating a multiple histogram formulation, the collection of distributions can be visualized alternatively as a multichannel feature image at a resolution lower than the original input data. To help ground intuition, Fig. 2.1 shows a frame from a video sequence along with a pictorial multichannel feature representation. The features are spatially divided into quadrants using a collection of four kernels. Histogram accumulation is performed according to (2.5), which subsequently produces four histograms (one histogram per quadrant) that can alternately be organized as a 2×2 image of multichannel features. In the limit, if just a single kernel were considered, a 1×1 multichannel feature image would be obtained, which is equivalent to a single histogram.

Given the preceding analysis, it appears that the notion of scale, as it concerns the image resolution over which features are accumulated (i.e., outer scale [25]) may become important in developing the proposed theoretical framework. With an eye toward subsequent derivations, minor additions will

be made to the histogram definitions in order to explicitly capture the notion of scale or level of image resolution. In particular, superscripts, l, are added to the histograms, the warp parameters, and the spatial coordinates to ensure all variables are operating at the same image resolution, as defined in terms of coupled kernel accumulation and spatial sampling. In doing so, $l = 0$ denotes the original image resolution and increasing values of l correspond to coarser resolutions. Incorporating this change yields a modified template histogram definition

$$\hat{q}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right) = C_q \sum_{j=1}^{J}\left(K_0^l\left(\mathbf{x}_j^0 - \bar{\mathbf{x}}_{\bar{n}}^0\right) Q^0\left(\mathbf{x}_j^0, m\right)\right), \tag{2.6}$$

where the normalization term is defined as

$$C_q = \frac{1}{\sum_{m_i=1}^{M}\sum_{j=1}^{J}\left(K_0^l\left(\mathbf{x}_j^0 - \bar{\mathbf{x}}_{\bar{n}}^0\right) Q^0\left(\mathbf{x}_j^0, m_i\right)\right)} \tag{2.7}$$

and Q^0 represents the multichannel features at their original scale. Additionally, $\mathbf{x}_j^0$ denotes pixels at full resolution, $\bar{\mathbf{x}}_{\bar{n}}^0$ is the center of the $\bar{n}^{\text{th}}$ kernel at the 0^{th} level and the kernel K_0^l represents the input and output resolutions with its subscript and superscript, respectively. The candidate histograms, (2.5), are analogously updated to include resolution of analysis, l.

Before proceeding with the definition of an error function for the multiple kernel histogram tracker, one final point should be noted. Rather than obtaining a warped image at a level, l, by warping the kernel, K, at full resolution and applying the result to the full resolution image, one can apply an unwarped kernel to the input image at full resolution and subsequently perform the analogous warp at level l (i.e., at the lower resolution). This equivalence is expressed as

$$\begin{aligned}\hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l, t\right) &\equiv \hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l, \mathbf{a}^l, t\right) \\ &= C_p \sum_{j=1}^{J}\left(K_0^l\left(\mathbf{x}_j^0 - \left(\bar{\mathbf{x}}_{\bar{n}}^0 + \mathsf{W}^0\left(\bar{\mathbf{x}}_{\bar{n}}^0\right)\mathbf{a}^0\right)\right) P^0\left(\mathbf{x}_j^0, m, t\right)\right),\end{aligned} \tag{2.8}$$

with superscripts on W, the warp matrix, and $\mathbf{a}$, the warp parameters, denoting the resolution at which the parametric warp is applied. This equivalence relation need not be expressed for the template, as it is never explicitly warped.

Now, the tracking error between target template, (2.6), and candidate, (2.8), that encompasses multichannel features can be defined in terms of SSD

as

$$\Omega\left(\mathbf{a}^l\right) = \sum_{\bar{n}=1}^{\bar{N}}\sum_{m=1}^{M}\left(\hat{q}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right) - \hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l, t\right)\right)^2, \tag{2.9}$$

where $\bar{N}$ is the number of kernels, M is the number of histogram bins (equivalently, feature channels), and $\bar{\mathbf{x}}_{\bar{n}}^l$ are the kernel centers. To provide a more compact notation, the temporal variable, t, will be suppressed when the meaning is unambiguous in the following. Adding a small perturbation to the warping parameters in (2.9) and performing a first-order Taylor series expansion results in

$$\begin{aligned}\Omega\left(\mathbf{a}^l + \Delta\mathbf{a}^l\right) = \sum_{\bar{n}=1}^{\bar{N}}\sum_{m=1}^{M}\Bigg[&\hat{q}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right) - \hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right) \\ &- \frac{\partial}{\partial\mathbf{a}^l}\left(\hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)\Delta\mathbf{a}^l\Bigg]^2.\end{aligned} \tag{2.10}$$

To compute the derivative of a candidate histogram, $\hat{p}_m^l$, the normalization variable, C_p, must be explicitly reintroduced, as it is also a function of $\mathbf{a}^l$, yielding

$$\begin{aligned}&\frac{\partial}{\partial\mathbf{a}^l}\left(\hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right) = \frac{\partial}{\partial\mathbf{a}^l}\left[\frac{p_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)}{\sum_{m_i=1}^{M} p_{m_i}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)}\right] = \\ &\left[\frac{\left(\left(\nabla_{\mathbf{x}^l} p_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)^\top \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\right)\left(\sum_{m_i=1}^{M} p_{m_i}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)}{\left(\sum_{m=1}^{M} p_{m_i}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)^2} - \right. \\ &\left.\frac{\left(\left(\sum_{m=1}^{M}\nabla_{\mathbf{x}^l} p_{m_i}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)^\top \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\right)\left(p_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)}{\left(\sum_{m_i=1}^{M} p_{m_i}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)^2}\right],\end{aligned} \tag{2.11}$$

where m_i ranges across all M channels, while $\nabla_{\mathbf{X}^l}$ is the gradient of the l^{th} level accumulated image with respect to the spatial parameters, x^l and y^l. The derivative, (2.11), will be denoted as $\bar{R}$ to provide a more compact, readable notation.

Reinserting $\bar{R}$ into (2.10) yields

$$\Omega\left(\mathbf{a}^l + \Delta\mathbf{a}^l\right) = \sum_{\bar{n}=1}^{\bar{N}}\sum_{m=1}^{M}\left(\hat{q}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right) - \hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l + \mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right) - \bar{R}\Delta\mathbf{a}^l\right)^2. \tag{2.12}$$

The derivative of the simplified error function, (2.12), is subsequently computed with respect to the motion parameter perturbations, $\Delta\mathbf{a}^l$, according to

$$\frac{\partial}{\partial \Delta\mathbf{a}^l}\left[\Omega\left(\mathbf{a}^l+\Delta\mathbf{a}^l\right)\right] = -2\sum_{\bar{n}=1}^{\bar{N}}\sum_{m=1}^{M}\bar{R}^\top\left(\hat{q}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)-\hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l+\mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)-\bar{R}\Delta\mathbf{a}^l\right). \tag{2.13}$$

Finally, setting (2.13) to zero so that the parameter perturbations can be isolated leads to an update equation for the motion parameters

$$\Delta\mathbf{a}^l = \left(\sum_{\bar{n}=1}^{\bar{N}}\sum_{m=1}^{M}\left[\bar{R}^\top\bar{R}\right]\right)^{-1}\sum_{\bar{n}=1}^{\bar{N}}\sum_{m=1}^{M}\left[\bar{R}^\top\left(\hat{q}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)-\hat{p}_m^l\left(\bar{\mathbf{x}}_{\bar{n}}^l+\mathsf{W}^l\left(\bar{\mathbf{x}}_{\bar{n}}^l\right)\mathbf{a}^l\right)\right)\right], \tag{2.14}$$

where $^{-1}$ denotes the matrix inverse. This update equation, (2.14), provides tracking as a spatial mapping (alignment) between the target template and candidate.

Multichannel template warp tracker. Analogous to the previous section, template warp tracking will be defined for *multichannel* image measurements (features), with P and Q denoting the target template and candidate images, resp. Again, the multichannel feature images are normalized to ensure appropriately weighted contributions from each channel [26], which also keeps the formulation consistent with that of multiple kernel histograms specified above. Thus, the normalized template and candidate feature images are defined as

$$\hat{Q}^l\left(\mathbf{x}^l,m\right) = \frac{Q^l\left(\mathbf{x}^l,m\right)}{\sum_{m_i=1}^{M}Q^l\left(\mathbf{x}^l,m_i\right)} \tag{2.15}$$

and

$$\hat{P}^l\left(\mathbf{x}^l+\mathsf{W}^l\left(\mathbf{x}^l\right)\mathbf{a}^l,m,t\right) = \frac{P^l\left(\mathbf{x}^l+\mathsf{W}^l\left(\mathbf{x}^l\right)\mathbf{a}^l,m,t\right)}{\sum_{m_i=1}^{M}P^l\left(\mathbf{x}^l+\mathsf{W}^l\left(\mathbf{x}^l\right)\mathbf{a}^l,m_i,t\right)}, \tag{2.16}$$

respectively, where m indexes a particular channel of features and all other notation is analogous to that of the previous section.

As in the previous section, the definitions of normalized template and candidate images, (2.15) and (2.16), make use of the superscript, l, to denote image resolution. In this section, since arrays of standard images are being operated upon (as opposed to histograms), pyramid processing will

be invoked explicitly as a convenient representation for manipulating multiresolution data [27]; however, the resulting derivations are *not* dependent upon pyramid processing. Here, the multichannel feature images form image pyramids on a channel by channel basis. A so-called equivalent weighting function [27], G_0^l, where the subscript and superscript indicate the input and output level, respectively, can be used to compute the image feature pyramid at level l, Q^l, from the original feature image, Q^0, according to

$$\hat{Q}^l\left(\mathbf{x}_n^l, m\right) = \left[\sum_{j=1}^{J} G_0^l\left(\mathbf{x}_j^0 - \mathbf{x}_n^0\right)\hat{Q}^0\left(\mathbf{x}_j^0, m\right)\right]\downarrow_{2^l}, \tag{2.17}$$

where $\mathbf{x}_n^0$ is the center of convolution, $\mathbf{x}_n^l$ is the corresponding pixel in image $\hat{Q}^l$, $\downarrow_{2^l}$ indicates factor of 2^l spatial subsampling, j ranges such that all pixels within the support of the equivalent weighting kernel, G_0^l, are considered. (While factor of 2^l subsampling is used here for concreteness, alternatives could be used, typically chosen to avoid aliasing under the operative kernel, G_0^l.) Likewise, warped and downsampled normalized candidate images can be produced using the present notation and the multiresolution warping conventions introduced earlier, (2.8), to yield

$$\begin{aligned}\hat{P}^l\left(\mathbf{x}_j^l + \mathsf{W}^l\left(\mathbf{x}_i^l\right)\mathbf{a}^l, m\right) &\equiv \hat{P}^l\left(\mathbf{x}_n^l, \mathbf{a}^l, m\right) \\ &= \left[\sum_{j=1}^{J} G_0^l\left(\mathbf{x}_{\tilde{j}}^0 - \left(\mathbf{x}_n^0 + \mathsf{W}^0\left(\mathbf{x}_n^0\right)\mathbf{a}^0\right)\right)\hat{P}^0\left(\mathbf{x}_j^0, m\right)\right]\downarrow_{2^l}.\end{aligned} \tag{2.18}$$

With normalized, downsampled, and warped feature images in place, the template warp tracker of this section can be constructed. Recall that the multiple kernel histogram tracker presented previously utilized the SSD error metric. To establish explicit connections between the two paradigms, the metrics should be equal, yielding the following error equation for the template warp tracker

$$\Omega\left(\mathbf{a}^l\right) = \sum_{n=1}^{N}\sum_{m=1}^{M}\left(\hat{Q}^l\left(\mathbf{x}_n^l, m\right) - \hat{P}^l\left(\mathbf{x}_n^l + \mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l, m, t\right)\right)^2. \tag{2.19}$$

Consistent with the above developments for kernel tracking, a small perturbation is added to the warping parameters in (2.19), allowing for a first-order Taylor series expansion according to

$$\Omega\left(\mathbf{a}^l + \Delta\mathbf{a}^l\right) = \sum_{n=1}^{N}\sum_{m=1}^{M}\Big(\hat{Q}^l\left(\mathbf{x}_n^l, m\right) - \hat{P}^l\left(\mathbf{x}_n^l + \mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l, m\right) -$$

$$\frac{\partial}{\partial \mathbf{a}^l}\left(\hat{P}^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)\right)\Delta\mathbf{a}^l\Big)^2. \tag{2.20}$$

Note that the temporal variable, t, has been suppressed for compactness. To compute the derivative of the warped candidate feature image, expansion must be performed because the warp parameters appear both in the numerator and denominator of the candidate definition, (2.16). This expansion yields

$$\frac{\partial}{\partial \mathbf{a}^l}\left[\hat{P}^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)\right]=\frac{\partial}{\partial \mathbf{a}^l}\left[\frac{P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)}{\sum_{m_i=1}^M P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m_i\right)}\right]=$$
$$\left[\frac{\left(\left(\nabla_{\mathbf{x}^l}^\top P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)\right)\mathsf{W}^l\left(\mathbf{x}_n^l\right)\right)\left(\sum_{m_i=1}^M P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m_i\right)\right)}{\left(\sum_{m_i=1}^M P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m_i\right)\right)^2}-\right.$$
$$\left.\frac{\left(\left(\sum_{m_i=1}^M \nabla_{\mathbf{x}^l}^\top P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m_i\right)\right)\mathsf{W}^l\left(\mathbf{x}_n^l\right)\right)\left(P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)\right)}{\left(\sum_{m_i=1}^M P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m_i\right)\right)^2}\right], \tag{2.21}$$

where $\nabla_{\mathbf{x}^l}^\top P^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m_i\right)$ is the spatial gradient of the candidate.

In subsequent equations, the derivative, (2.21), will be denoted as R to provide a more compact, readable notation. Reinserting R into (2.20) yields

$$\Omega\left(\mathbf{a}^l+\Delta\mathbf{a}^l\right)=\sum_{n=1}^N\sum_{m=1}^M\left(\hat{Q}^l\left(\mathbf{x}_n^l,m\right)-\hat{P}^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)-R\Delta\mathbf{a}^l\right)^2. \tag{2.22}$$

The derivative of the simplified error function, (2.22), is next computed with respect to the motion parameter perturbations

$$\frac{\partial}{\partial\Delta\mathbf{a}^l}\left[\Omega\left(\mathbf{a}^l+\Delta\mathbf{a}^l\right)\right]=$$
$$-2\sum_{n=1}^N\sum_{m=1}^M R^\top\left(\hat{Q}^l\left(\mathbf{x}_n^l,m\right)-\hat{P}^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)-R\Delta\mathbf{a}^l\right). \tag{2.23}$$

Finally, setting (2.23) to zero and isolating the motion parameters results in

$$\Delta\mathbf{a}^l=\left(\sum_{n=1}^N\sum_{m=1}^M\left[R^\top R\right]\right)^{-1}\sum_{n=1}^N\sum_{m=1}^M\left[R^\top\left(\hat{Q}^l\left(\mathbf{x}_n^l,m\right)-\hat{P}^l\left(\mathbf{x}_n^l+\mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l,m\right)\right)\right], \tag{2.24}$$

which fully defines the template warping tracker for multichannel features.

Drawing equivalences. Comparison of the kernel histogram, (2.14), and template warp, (2.24), tracking update equations reveals that they have the same general form, but slight notational discrepancies. It will now be shown that these differences are purely superficial. To draw this connection, a two step approach is taken. First, the summations across both update equations will be demonstrated to perform the same series of operations when the systems are properly configured. Second, it will be shown that under mild conditions, the processed feature data (multiple kernel histograms and multichannel feature images) are equivalent.

With respect to the first step, it is straightforward to argue that the inner summations, denoted as $\sum_{m=1}^{M}$ in both equations, are equivalent. These two summations both accumulate error over all M feature channels. Additionally, a connection must be drawn between the outer summation in (2.14) over $\bar{N}$ kernels and that in (2.24) over N pixels. This connection is formed by considering the relationship between kernel histogram accumulation and pyramid downsampling.

An illuminating example comparing the process of placing kernels within a multichannel feature image and downsampling such an image is shown in Fig. 2.2. On the left most panel, an input cube of multichannel features is shown. The x and y axes indicate the spatial dimensions; whereas, the depth of the cube denotes the feature channels. This *multichannel* feature image is the primary input for subsequent processing for both the multiple kernel histogram tracker and the template warp tracker. The middle column of the figure illustrates how the multichannel feature image is processed by the multiple kernel histogram tracker. Specifically, the tracker is configured to place kernels on a uniform grid with a pixel spacing of two, where "K" indicates a kernel center. After kernel application, the result can be visualized as a set of images, with spatial dimensions of 4×4. The right-most column of the figure illustrates the downsampling procedure for the template warp tracker. Following appropriate blurring, downsampling is performed over a single level (from Level 0 to Level 1) and pixels are subsampled by a factor of two along each spatial dimension, with retained pixels indicated by red arrows. After downsampling, the template warping tracker will operate on a set of 4×4 downsampled feature images. Notably, the pixels that are explicitly retained and subsequently processed by both trackers originate from the same pixels at full resolution. This discussion shows that the summations in the update equations, (2.14) and (2.24), can be constructed to perform

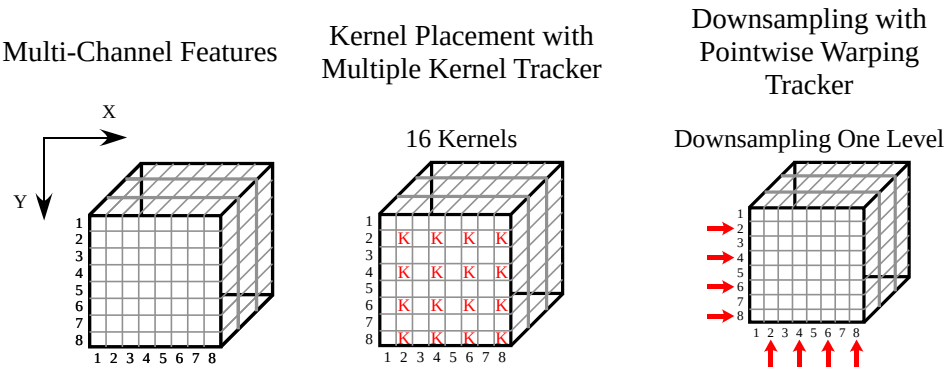

Figure 2.2: Comparison of kernel accumulation vs. downsampling. See text for details.

identical accumulations. Mathematically, the number of kernels is set equal to the resolution of the l^{th} level image where tracking is performed, i.e., $\bar{N} = N$ and the variables of summation $\bar{n}$ and n are configured to consider equivalent pixels on a regularly spaced grid. Under such a configuration, the summations for both tracking paradigms perform the same series of operations, $\sum_{\bar{n}=1}^{\bar{N}} \equiv \sum_{n=1}^{N}$, establishing that the accumulations in both update equations, (2.14) and (2.24), can be designed to be equivalent.

To continue drawing connections between the two formulations, the kernel histograms, (2.6), and normalized multichannel feature images, (2.17), must also be shown to be two different conceptualizations of the same data. A side by side comparison reveals that these two expressions share many similarities. Both equations perform normalization due to its benefits when manipulating multichannel data [26], i.e., C_q in (2.6) and normalized images, $\hat{Q}$, in (2.17), such that pointwise summations across channels are unity. The template image, (2.17), includes explicit notation indicating that subsampling is performed across each pyramid level, $\downarrow_{2^l}$. However, it was just shown that equivalent subsampling can be performed in the multiple kernel histogram framework by appropriately selecting the kernel positions.

Another apparent difference between the two representations is that in (2.6) pixels in the template image are weighted by a spatial kernel; whereas, in (2.17) the image is convolved with a blurring filter. If the kernel weighting function, K_0^l, and the blurring filter, G_0^l, are equal, these two equations will perform the same series of operations and will thus be equivalent. Gaussian filters are often used for blurring during downsampling [27], while Gaussian

spatial kernels have been used previously in kernel histogram trackers, e.g., [2]. Thus, a Gaussian kernel has proven to be an effective choice in both paradigms, but for equivalence between (2.6) and (2.17), the only critical requirement is that $K_0^l = G_0^l$. An analogous argument can be presented for the warped candidate data, (2.8) and (2.18).

By drawing this connection between the histogram data and the feature images, the two step analysis is complete. The primary point to emphasize is as follows: When the update equations for the multiple histogram and the template warping systems are configured in an analogous fashion, they both produce an identical tracker. As a corollary, either tracker formulation can be used to describe a wide range of systems, extending from a single kernel histogram instantiation to a traditional template warp tracker. Thus, the two formulations can be used interchangeably to describe a continuum of region trackers.

Unifying framework. This section presents a generalized notation describing the continuum of trackers from single kernel histogram to template warping. By changing the parameter settings of this generalized equation, a tracker that behaves identically to a single kernel histogram tracker, a standard template warp tracker, or at any point in between can be instantiated. The primary variable that determines the location at which a tracker appears along the continuum is the *spatial resolution* of analysis, l, which captures the amount of retained spatial layout information. This spatial arrangement information is dependent upon the sampling density of the measurements and the size of the aggregation region. To provide as broad a framework as possible, generalization can be performed at the level of the error function, since the previous section showed that summations and data (feature) representations in the kernel histogram, (2.9), and template warping trackers, (2.19) can be made identical.

Accordingly, a generalized error function to define region trackers across the continuum is given by

$$\Omega\left(\mathbf{a}^l\right) \;=\; \sum_{n=1}^{N}\sum_{m=1}^{M} \xi\left(\hat{Q}^l\left(\mathbf{x}_n^l, m\right), \hat{P}^l\left(\mathbf{x}_n^l + \mathsf{W}^l\left(\mathbf{x}_n^l\right)\mathbf{a}^l, m, t\right)\right), \quad (2.25)$$

where ξ is a general error function that operates on multichannel image features ($\hat{P}^l$ and $\hat{Q}^l$). A simple error function, SSD, was utilized in the previous subsection; however, a more sophisticated function such as a robust estimator [18] can be alternatively incorporated. The optimization strategy for locating extrema of (2.25) with respect to the motion parameters, $\mathbf{a}$, is

another aspect that the tracker designer can control and may depend on the selected error metric. In previous subsections, this paper presented a gradient-based approach for computing the parameter updates. Alternative optimization methods may include simple "spotting" (i.e., exhaustive search) approaches [14] or Newton-like techniques [17]. Finally, the particulars of kernel placement and support can be specified by the designer.

Chapter 3

Empirical results

This section presents example instantiations of the developed general region tracker formulation to uncover advantages and disadvantages of including varying amounts of spatial layout in the target representation. It should be stressed that the current evaluation is *not* concerned with comparing a new state-of-the-art tracker against other strong trackers, rather with the coverage of a range of extant trackers. The instantiations considered in this evaluation will be referred to in terms of their spatial resolution, which captures both the spatial support over which local feature measurements are aggregated and the number of kernels that are placed. Analogous to pyramid processing nomenclature [27], resolution will be specified in terms of levels, **Level 0** ... **Level L**, where the former corresponds to a traditional template warping tracker (single pixel impulse kernel at every point) and the latter is analogous to a single kernel histogram tracker (kernel covers the entire target support). Thus, the set of trackers considered encompasses systems that are comparable to several standard algorithms in the literature, including mean shift [2] (**Level L**), *FragTrack* [14] ($0 <$ **Level l** $<$ L), and template warping [23] (**Level 0**).

In the evaluation, intermediate levels, $0 <$ **Level l** $<$ L, are derived recursively from Level 0 via factor of two reduction in the number of kernels along both spatial axes with matched increase in aggregation using a Gaussian kernel. While the framework set forth in the previous section allows for arbitrary kernel placement and support, the factor of two variation provides systematic exploration of the range between single kernel and template warp tracking.

For all trackers in all experiments, a common set of parameters was em-

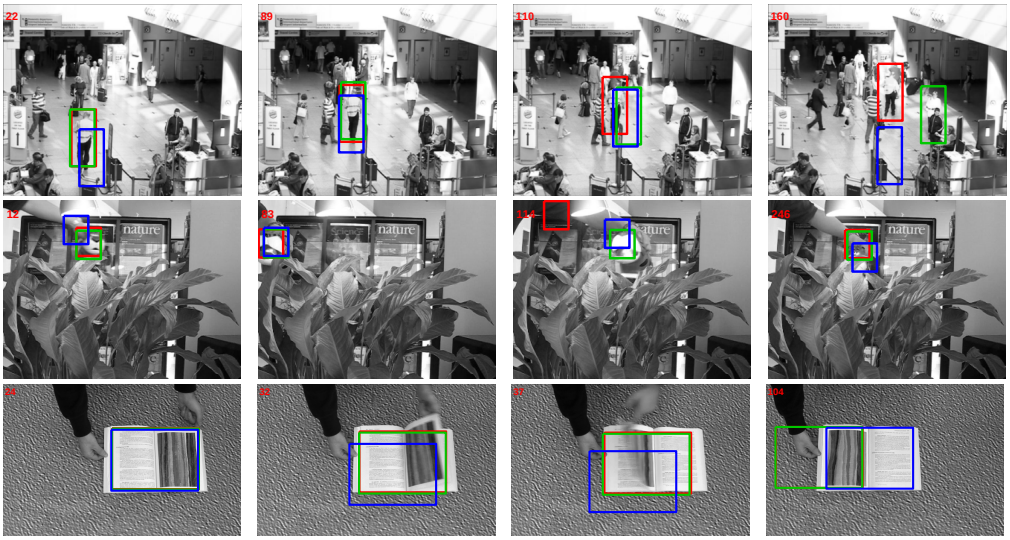

Figure 3.1: Qualitative tracking comparisons. Frame numbers are shown in the top left corner. Top-to-bottom by row are *PETS2007*, *Tiger2*, and *PageFlip*. Boxes correspond to the tracker with a kernel at every pixel (red), the Level 2 tracker, downsampled twice (green), and the single kernel tracker (blue), respectively.

ployed. For multichannel features, 3D, xyt, spatiotemporal oriented energies derived from pointwise application of quadrature pair Gaussian second derivative filters and their Hilbert transforms to full resolution imagery are used [28], as they capture both spatial and temporal target characteristics and have been shown to yield strong performance at both ends of the proposed continuum [21]. The error function for evaluating target candidates was taken as SSD. To optimize the error function, spotting, i.e., exhaustive search over translation, was performed over a 7×7 pixel search grid, which is roughly twice the maximum interframe displacement of a target in the videos considered. Finally, a simple autoregressive template update mechanism is employed in all trackers [29].

Three challenging video sequences are considered for evaluation; see Fig. 3.1. To initiate tracking in each video, an outline of the target was provided manually in the first frame. The first example, *PETS2007* [30] (Fig. 3.1, top row), illustrates the challenge of tracking a single target (a person) against similarly appearing background clutter (other people), including mu-

tual occlusions. Here, it is seen that maintaining additional spatial layout yields better performance, as tracking accuracy incrementally improves as one moves from single kernel through an intermediate number of kernels to a kernel at every pixel. These results can be explained by the fact that the background clutter shares a very similar appearance to the target and is often found in close proximity (even momentarily occluding the target); thus, maintenance of increased amounts of distinguishing spatial information yields corresponding success. The greater number of kernels provides most improvement near the end of the sequence, when clutter is greatest. The second example, *Tiger2* [31] (Fig. 3.1, middle row), shows the challenge of tracking a deforming object (toy tiger with articulating mouth) moving rapidly amongst clutter (a plant). Here, it is seen that maintaining an intermediate number of kernels yields best performance. This result can be explained by the single kernel instantiation still not maintaining a representation that is rich enough to distinguish the target from the clutter, while the pixelwise version is not allowing for sufficient deformation. In contrast, the intermediate representation provides a good tradeoff in these requirements. The third example, *PageFlip* (Fig. 3.1, bottom row, author's video), shows the case of tracking through a radical appearance change, as a book translates through the scene and a page flip occurs mid-way. In this situation, only the single kernel case can track accurately through the page flip. By collapsing the local feature measurements across target support, this tracker capitalizes on the fact that while the local appearance of the target changes drastically, its global attributes remain consistent: The open book remains characterized by a page of text and painting reproduction, even though they are not identical. The other instantiations are too sensitive to local appearance to succeed in the presence of this challenge.

A quantitative analysis of tracking performance was also completed using ground truth target bounding boxes that were obtained by hand at a minimum of every fifth frame for each video. To summarize the results, the center of mass Euclidean pixel distance error was averaged across all frames for a dense sampling of tracking levels, yielding the statistics in Table 3.1. These results provide quantitative support for the qualitative trends that have been described.

While these experiments are *not* meant to be exhaustive, they provide evidence regarding the advantages and disadvantages of maintaining various amounts of spatial layout information with respect to the challenges present in the videos under analysis. These initial experiments support the

Table 3.1: Summary of quantitative results for tracker instantiations. Values listed are pixel distance errors for the center of mass points averaged over all frames. Green and red show best and worst performance, resp. NA indicates that given the target size, the single kernel case already has been achieved at an earlier level.

Level	0	1	2	3	4	5	6	7
PETS2007	**5.4**	11.6	21.6	14.7	21.5	**47.5**	NA	NA
Tiger2	**43.7**	20.5	**19.5**	21.0	26.5	NA	NA	NA
PageFlip	82.5	83.3	83.7	84.6	84.0	**87.6**	69.2	**16.6**

following observations regarding general performance trends: Maintenance of maximally dense spatial representations provides a high degree of target discriminability against clutter (e.g., *PETS2007*); however, this comes at the price of being overly sensitive to target deformations that are better ignored (e.g., *Tiger2, PageFlip*). Radical appearance changes (e.g., *PageFlip*) can be dealt with best by maintaining only a coarse target layout. Moderate degrees of deformation, especially in the presence of clutter (e.g., *Tiger2*) are accommodated well by maintaining an intermediate amount of target layout. The framework of Sec. 3 provides a systematic approach to exposing and understanding the presented range of phenomena and can guide further investigation.

Chapter 4

Discussion

This paper has presented a unifying framework that creates a continuum of region trackers spanning from single kernel histogram to template warping systems. A detailed formal analysis has shown how this range of trackers can be related in terms of the amount of maintained spatial organization in their target representations and thereby highlights their commonalities and differences. Leveraging this unifying framework, a systematic empirical exploration of numerous trackers instantiated from this framework was performed. This study revealed tradeoffs in tracking performance as the amount of spatial organization information was varied.

Future research can exploit the proposed framework in a variety of ways. From a theoretical perspective, it would be of interest to expand the framework to encompass an even wider range of trackers (e.g., blob trackers) [1]. From a design perspective, it would be of interest to leverage the framework in the development of novel, state-of-the-art algorithms (e.g., trackers that automatically choose an appropriate amount of spatial information to include, based on the tracking task/data at hand). From an empirical perspective, the framework can guide the development of experiments that appropriately tax a range of operating points along the continuum of spatial organization. More generally, the presented research provides twofold benefit to the field by unifying a wide range of extant region trackers and providing a mechanism for evaluating the importance of spatial layout in a particular tracking task in a principled fashion.

Bibliography

[1] Yilmaz, A., Javed, O., Shah, M.: Object tracking. Comp. Surv. **38** (2006) 1–45

[2] Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR. (2000) 142–149

[3] Nummiaro, K., Koller-Meier, E., Gool, L.V.: An adaptive color-based particle filter. IVC **21** (2002) 99–110

[4] Okuma, K., Taleghani, A., Freitas, N.D., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: ECCV. (2004) 28–39

[5] Hager, G., Dewan, M., Stewart, C.: Multiple kernel tracking with SSD. In: CVPR. (2004) 790–797

[6] Dewan, M., Hager, G.D.: Toward optimal kernel-based tracking. In: CVPR. (2006)

[7] Parameswaran, V., Ramesh, V., Zoghlami, I.: Tunable kernels for tracking. In: CVPR. (2006) 2179–2186

[8] Babu, R.V., Perez, P., Bouthemy, P.: Robust tracking with motion estimation and local kernel-based color modeling. IVC **25** (2007) 1205–1216

[9] Fan, Z., Yang, M., Wu, Y.: Multiple collaborative kernel tracking. PAMI **29**(7) (2007) 1268–1273

[10] Backhouse, A., Khan, Z.H., Gu, I.Y.: Robust object tracking using particle filters and multi-region mean shift. In: PCM. (2009) 393–403

[11] Kang, J., Cohen, I., Medioni, G.: Continuous tracking within and across camera streams. In: CVPR. (2003) 267–272

[12] Wang, F., Yu, S., Yang, J.: A novel fragments-based tracking algorithm using mean shift. In: ICARCV. (2008) 694–698

[13] Kwon, J., Lee, K.M.: Tracking of a non-rigid object via patch-based dynamic appearance modeling and Monte Carlo sampling. In: CVPR. (2009) 1208–1215

[14] Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR. (2006) 798–805

[15] Guskov, I.: Kernel-based template alignment. In: CVPR. (2006)

[16] Lucas, B., Kanade, T.: An iterative image registration technique with application to stereo vision. In: DARPA IUW. (1981) 121–130

[17] Bergen, J.R., Anandan, P., Hanna, K.J., Hingorani, R.: Hierarchical model-based motion estimation. In: ECCV. (1992) 237–252

[18] Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. CVIU **63** (1996) 75–104

[19] Jepson, A., Fleet, D., El-Maraghi, T.: Robust online appearance models for visual tracking. PAMI **25** (2003) 1296–1311

[20] Ross, D.A., Lim, J., Lin, R.S.: Incremental learning for robust visual tracking. IJCV **77** (2008) 125–141

[21] Cannons, K.J., Gryn, J.M., Wildes, R.P.: Visual tracking using a pixelwise spatiotemporal oriented energy representation. In: ECCV. (2010) 511–524

[22] Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: CVPR. (2012)

[23] Baker, S., Matthews, I.: Lucas-Kanade 20 years on. IJCV **56** (2004) 221–255

[24] Cannons, K., Wildes, R.: Spatiotemporal oriented energy features for visual tracking. In: ACCV. (2007) 532–543

[25] Koenderink, J.J.: The structure of images. Bio. Cyb. **50** (1984) 363–370

[26] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley (2000)

[27] Burt, P., Adelson, E.: The Laplacian pyramid as a compact image code. IEEE TC **31** (1983) 532–540

[28] Freeman, W., Adelson, E.: The design and use of steerable filters. PAMI **13** (1991) 891–906

[29] Irani, M., Rousso, B., Peleg, S.: Computing occluding and transparent motions. IJCV **12** (1994) 5–16

[30] PETS. http://pets2007.net/ (2007)

[31] Babenko, B., Yang, M., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR. (2009) 983–990