Exploring Weak Stabilization for Motion Feature Extraction

Dennis Park
UC Irvine
iypark@ics.uci.edu

C. Lawrence Zitnick
Microsoft Research
larryz@microsoft.com

Deva Ramanan
UC Irvine
dramanan@ics.uci.edu

Piotr Dollár
Microsoft Research
pdollar@microsoft.com

Abstract

We describe novel but simple motion features for the problem of detecting objects in video sequences. Previous approaches either compute optical flow or temporal differences on video frame pairs with various assumptions about stabilization. We describe a combined approach that uses coarse-scale flow and fine-scale temporal difference features. Our approach performs weak motion stabilization by factoring out camera motion and coarse object motion while preserving nonrigid motions that serve as useful cues for recognition. We show results for pedestrian detection and human pose estimation in video sequences, achieving state-of-the-art results in both. In particular, given a fixed detection rate our method achieves a five-fold reduction in false positives over prior art on the Caltech Pedestrian benchmark. Finally, we perform extensive diagnostic experiments to reveal what aspects of our system are crucial for good performance. Proper stabilization, long time-scale features, and proper normalization are all critical.

1. Introduction

Object detection is a central task in vision. Most approaches have focused on the static-image setting; indeed, a common method for detecting objects in video is to run an image-based detector on each frame. Significant progress has been made in static-image object detection over the past few years, in large part due to the improvement of low-level features coupled with classifiers such as SVMs [7] and boosting [26]. In this work, we explore the motion counterpart for object detection in video. We show that one can exploit simple motion features to significantly increase detection accuracy with little additional computation.

Image motion observed in videos is the result of several sources, Figure 1. We classify image motion into three types using a stationary world coordinate frame and a moving object coordinate frame. Camera-centric motion is the movement of the camera with respect to the world. Object-centric motion is the movement of the object centroid with respect to the world. Finally, part-centric motion is the movement of object parts with respect to the object. These three types of motion provide different cues for recognition.

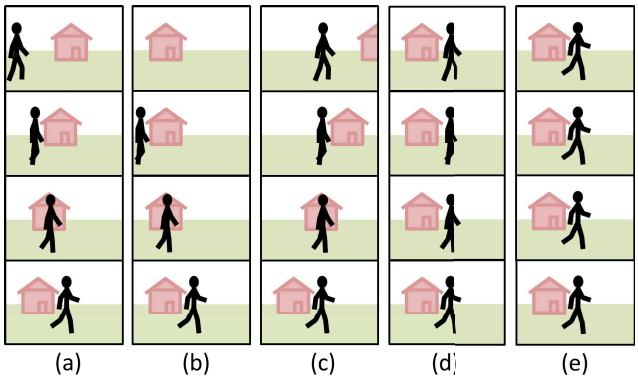

Figure 1: Illustration of various types of video stabilization: (a) no stabilization, (b) camera motion stabilization, (c) object-centric motion stabilization, (d) camera and object-centric motion stabilization, and (e) full stabilization of camera, object-centric, and part-centric motion. We posit that for detecting articulated objects such as people the majority of useful motion information is contained in part-centric motion. We therefore attempt to stabilize both camera and object-centric motion, as in (d).

Prior work makes different assumptions about which motion types are useful versus nuisance factors. A simple approach is to directly compute image motion features on raw video. In this case, the observed image motion contains camera-, object-, and part-centric motion. Methods that define motion features using optical flow or spacetime gradients often take this route [29]. One can partly remove camera motion by looking at differences of flow [8]. A more direct approach is to simply compute motion features on a stationary camera, such as [27]. Such motion features encode both object- and part-centric motion. The large body of techniques that rely on background subtraction take this approach [24]. When the camera is moving, one may try to register frames using a homography or egomotion estimation [18, 19], which removes some camera-centric motion but can be challenging for dynamic scenes or those with complex 3D geometry. Finally, other techniques compute optical flow in an object-centric coordinate frame [13]; Figure 1(c) shows that such an approach actually encodes both camera- and part-centric motion.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.371

In this paper, we posit (and verify by experiment) that the majority of useful motion information for detecting articulated objects such as people is contained in part-centric motion. As shown in Figure 1, there are numerous types of video stabilization. To allow the temporal features to easily extract part-centric motion information, we attempt to stabilize both camera and object-centric motion, Figure 1(d). We accomplish this by using *coarse-scale* optical flow to align a sequence of image frames. Weak stabilization using coarse-scale flow has the benefit of aligning large objects such as the background or a person's body without removing detailed motion such as an object's parts, Figure 1(d,e). While artifacts may exists around large flow discontinuities, we demonstrate that coarse-scale flow is robust in practice.

We use temporal difference features to capture the part-centric motion that remains after weak stabilization. While features based on fine-scale optical flow [13, 8] may be extracted from the stabilized frames, fine-scale flow is notoriously difficult to extract for small parts such as arms [4]. We demonstrate that when sampled at the proper temporal intervals, simple temporal difference features are an effective alternative capable of achieving state-of-the-art results.

We perform a thorough evaluation of motion features for object detection in video. We focus on detecting pedestrians in moving cameras [12] as well as pose estimation from static cameras [1]. We demonstrate significant improvements from integrating our motion features into three distinct approaches: rigid SVM detectors defined on HOG features [7], articulated part models defined on HOG features [16, 31], and boosted detectors defined on channel features [11]. Notably, we report a five-fold reduction in false positives at fixed detection rates on the Caltech Pedestrian Benchmark, a significant improvement over prior art.

Finally, we perform an exhaustive sweep over various parameter settings of our model to analyze what aspects are important. We find it crucial to (1) compute optical-flow at the right level of coarseness to provide camera and object-centric stabilization, (2) compute difference features over long time scales (because motion over pairs of successive frames may be too subtle to measure) and (3) normalize motion features appropriately for use with linear SVMs.

2. Related Work

Optical-flow-based features: A popular strategy for video-based recognition is to extend static image features into the temporal domain through use of optical flow. Examples include spatially blurred flow fields [13] or histograms of optical flow vectors [8, 28]. In particular, Dalal *et al*. explored flow-augmented versions of their HOG descriptor [8] termed histograms of flow (HOF). Although HOF performed well for classification, Dalal's thesis admitted that it under-performed a HOG template when evaluated for detection [6]. Walk *et al*. [28] proposed a number of modifications to the HOF features that resulted in modest gains in detection performance. Recognizing the difficulty of accurate fine-scale flow estimation (aperture problem, singularities, etc.), [25] proposed directly comparing motion fields without explicit computation of flows.

Temporal-difference features: Temporal differencing, or temporal gradient features, date back to the early work of [2]. Since two-frame differencing might be too weak to produce a signal for slow-moving objects, [5, 20] describe approaches for multi-frame differencing. A related and popular approach is histograms of spacetime gradients [32, 21]. For stationary cameras, temporal difference features can be computed on background models, yielding background-subtraction masks [24]. Our approach can be seen as a combination of optical-flow and temporal differencing as we compute differences on spacetime windows that are weakly-stabilized with coarse optical flow.

Action classification: Many of the above motion features have been explored in the context of action classification [10, 21]. In particular, [29] performs a thorough evaluation of motion descriptors, discovering that histograms of flow perform well. For our setting of detecting low resolution objects in videos, traditional flow fails because small movements are difficult to estimate reliably. While effective for behavior classification, space-time interest points have not proven useful for object detection.

Tracking: An alternate use of temporal information to improve detection reliability is to explicitly track objects. For example, detection may be improved by tracking repeated detections [3]. Most trackers tend to define motion models on static image features, although exceptions do exist [15]. Impressive results have also been shown on a system wide integration of detectors [14, 30, 17]. Such approaches are orthogonal to ours as we aim to improve the quality of the detections themselves through use of more informative image features.

3. Approach

In this section, we describe our basic approach to motion feature extraction. We begin by discussing basic notation and static features. We then describe our approach to weakly-stabilizing video frames and our resulting motion features. Results are provided in the following section.

Notation: Let $\mathcal{I}_t$ denotes the t-th frame of a given video and I_t denote an image patch from $\mathcal{I}_t$. The spatial extent of I_t in the frame is implicitly defined by the detection task. For pedestrian detection, I_t is a fixed-size 32×64 pixel patch. To detect people at different scales we use an efficiently computed image pyramid [9].

Static features: In addition to the motion features introduced below, we use one of two sets of static features densely computed on the *current* frame. Our first set of static features are the *channel features* described in [11]. As in [11], our channels include color (3 channels), gradient magnitude (1 channel) and gradient quantized by orien-

(a) Raw (b) σ=4 (c) σ=32

Figure 2: **Stabilization using coarse-scale LK flows.** We show temporally distant 3-frame sequences stabilized onto the last frame (bottom row). The red box in each frame is the location of the person in the last frame. (**a**) In the raw video, the person shifts from left to right due to camera and object motion. (**b**) Using fine-scale LK flows, the overall body is stabilized onto the last frame at the cost of distortion in body parts (most visible at the heads and legs of the top row). (**c**) Using coarse-scale LK flows the warped images are aligned in terms of the overall body location while still preserving clear motions of body parts.

tation (6 channels). Our second type of static features is the commonly used Histogram of Oriented Gradients (HOG) descriptor [7]. Specifically, we compute histograms of gradients using 9 orientations on an 8×16 grid of 4×4 cells.

3.1. Stabilizing videos

Our goal is to compute motion features based on part-centric motion, such as the movement of a person's limbs. This requires weakly stabilizing image frames to remove both camera and object-centric motion while preserving the part-centric motion. We accomplish this by using *coarse-scale* optical flow to align a sequence of frames.

We estimate optical flow using the approach of Lucas-Kanade [22] but applied in a somewhat non-standard manner. Lucas-Kanade proposed a differential approach to flow

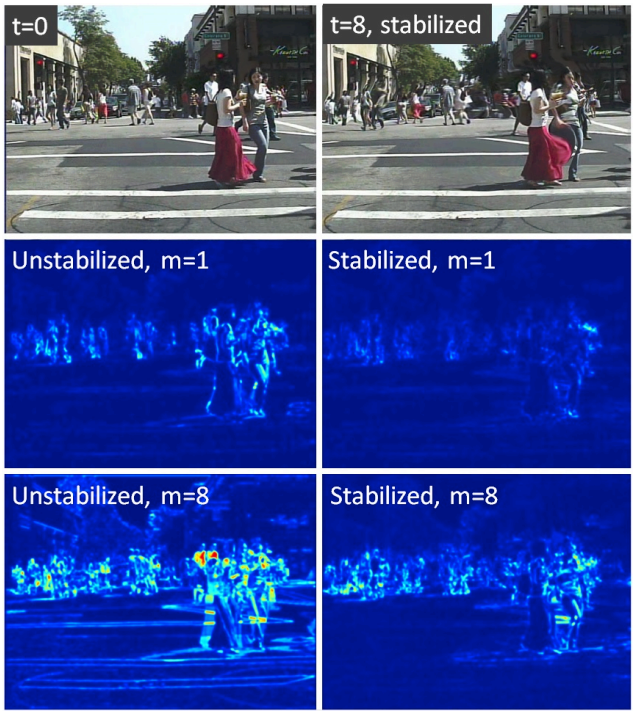

Figure 3: Example temporal frame differences using unstabilized and weakly stabilized frames spaced one frame apart ($m = 1$) and 8 frames apart ($m = 8$). When $m = 1$ there exists minimal temporal information. With larger frame spans ($m = 8$) temporal differences appear. However, weak stabilization is needed to remove non-informative differences resulting from camera and object motion.

estimation that is commonly implemented hierarchically. A window radius σ controls the scale of the flow. Typically, σ must be large enough to provide reliable local flow estimate but small enough to capture fine motions. Instead, *coarse* flow can be computed using a large radius σ. This offers dual advantages: the flow estimates are both more reliable and faster to compute.

We compute Lucas-Kanade flows with σ typically ranging from 8 to 32 pixels (16×16 to 64×64 windows). We denote the computed flow field from frame $\mathcal{I}_t$ to frame $\mathcal{I}_{t-1}$ as $W_{t,t-1}$. $\mathcal{I}_{t-1,t}$ is frame $\mathcal{I}_{t-1}$ warped to frame $\mathcal{I}_t$ using the flow field $W_{t,t-1}$. We write an image patch from the warped image as $I_{t-1,t}$. In practice, we find $W_{t,t-1}$ stabilizes the majority of motion due to camera and object-centric motion, as shown in Figure 2. Computing the coarse flows is fast (no need to compute flow at finest scale) and fairly robust (due to the large σ).

When stabilizing across multiple frames, we compute the global motion $W_{t,t-n}$ by progressively warping and summing pairwise flow fields. We found this to work better in practice than computing the potentially large flow directly between frames $\mathcal{I}_t$ and $\mathcal{I}_{t-n}$.

3.2. Motion features

Given (weakly) stabilized image frames, we propose the use of simple temporal differencing or temporal gradient features. We now describe the numerous variants that we experimentally evaluate. The temporal gradient is defined as the difference between two frames,

$$D^{\sigma} = I_t - I_{t-1,t}, \tag{1}$$

where σ is the scale of the computed flow. Because σ is tuned to be roughly the size of an object, we expect the temporal gradient to contain useful cues about nonrigid object motion that are helpful for detection, as in Figure 3. We denote temporal gradient on unstabilized frames as D^{US}:

$$D^{US} = I_t - I_{t-1} \tag{2}$$

Using multiple frames: We previously defined the difference features over pairs of frames. In many instances, the amount of motion observed between subsequent frames may be quite small, especially with slow moving objects. Consider Figure 2; it is hard to see the difference in poses between temporally adjacent frames. We alleviate this by considering multiple frames, or frames spaced further apart. Next, we define a family of multi-frame approaches.

First, we consider the simple approach of computing multiple frame differences between the current frame and $k = n/m$ other frames spaced apart temporally by m frames from $t - m$ to $t - n$. We refer to m as the *frame skip* and n as the *frame span*.

$$D_0^{\sigma}(n,m) = \begin{bmatrix} I_t - I_{t-1m,t} \\ I_t - I_{t-2m,t} \\ \vdots \\ I_t - I_{t-km,t} \end{bmatrix} \tag{3}$$

Using this notation, D^{σ} in Equation (1) computed from only two neighboring frames is equivalent to $D_0^{\sigma}(1,1)$.

Another approach is to compute the set of differences between neighboring frames within a multiframe set,

$$D_1^{\sigma}(n,m) = \begin{bmatrix} I_t - I_{t-m,t} \\ I_{t-m,t} - I_{t-2m,t} \\ \vdots \\ I_{t-(n-m),t} - I_{t-n,t} \end{bmatrix} \tag{4}$$

Finally, we may also compute the difference between the mean frame M_t and the neighboring frames,

$$D_M^{\sigma}(n,m) = \begin{bmatrix} M_t - I_{t-0m,t} \\ M_t - I_{t-1m,t} \\ \vdots \\ M_t - I_{t-km,t} \end{bmatrix}, \tag{5}$$

where $M_t = \frac{1}{k+1} \sum_{i=0}^{k} I_{t-im,t}$

Rectified features: Previously, we defined our temporal difference features using the signed temporal gradient. Several other possibilities also exist for encoding the temporal differences, such as using the absolute value of the temporal gradient or using rectified gradients. Rectified gradients compute two features for each pixel's temporal gradient dt corresponding to $\max(0, dt)$ and $\max(0, -dt)$. The motivation for this is that the sign of the gradient might provide additional information for detection (e.g. people often have darker hair color or clothing than the background).

Feature pooling: To add a small amount of spatial invariance, all of our features are pooled over a $c \times c$ sized rectangular window. In all our experiments our pooling size is 4×4. The pooling is the same as for the static features.

Feature normalization: The contrast between a person to be detected and the background may vary due to lighting, background texture or clothing. This affects both static and temporal difference features. Static features such as HOG [7] account for this using feature normalization. We follow a similar approach, but extended to spatio-temporal blocks. After pooling our difference features over $c \times c$ neighborhoods, we construct overlapping $s \times s \times t$ blocks of cells with spatial extent $s = 2$ and temporal extent $t = 2$ (analogous to R-HOG, but extended in time). We then $L1$ normalize each block feature (which we found to outperform $L2$ normalization). To improve performance, we found it important to clip the computed $L1$ norm of each block to have a maximum value of .05. Finally, following the approach of [16], we use the average of eight normalized values (computed from overlapping spacetime blocks) as the final feature.

4. Experimental results

In this section, we present a thorough evaluation of the family of features described above. We evaluate our results on two datasets, the Caltech Pedestrian dataset [12] and the MindsEye dataset [1]. We begin by exploring the feature parameter space on the task of pedestrian detection using a boosting classifier [11]. For the use of linear SVM [7] classifiers we show that normalizing features is crucial. With the optimal setting, state-of-the-art results are shown using boosting and linear classifiers. We conclude our experimental results by showing promising results on the challenging task of part detection using the MindsEye dataset [1].

4.1. Pedestrian detection

In this section, we explore various parameter settings on the Caltech Pedestrian dataset [12], which consists of 10 hours of real-world video footage from a car-mounted camera. The full dataset contains over 350,000 pedestrian detections. As is recommended practice [12], we train and evaluate using every 30^{th} frame and a smaller "reasonable" subset of bounding boxes containing pedestrians 50 pixels or taller and with limited occlusion. For boosted classifiers, we average results over 20 trials with different random seeds to increase their statistical significance.

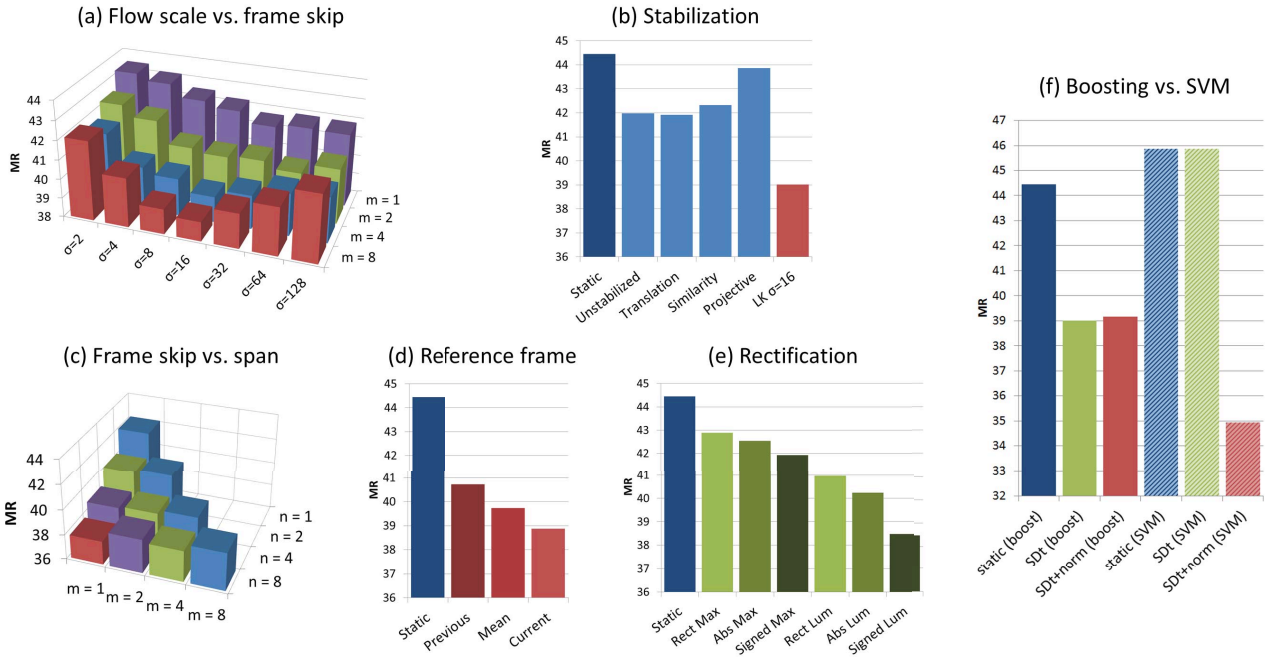

Figure 4: Results for various parameter sweeps on the Caltech pedestrian dataset. These include (a) adjusting the flow scale σ vs. the frame skip m, (b) other forms of stabilization, (c) frame skip m vs. frame span n, (d) various types of reference frames for computing $D(m, n)$, (e) different types of rectification for utilizing the color channels, and (f) boosting vs. SVM results with and without normalization. The best results, $D_0^{16}(8, 4)$, are achieved using $\sigma = 16$, $m = 4$, $n = 8$, the current frame as reference, and the signed temporal differences of the luminance channel. The SVM classifier outperforms the boosting classifier when normalization is used. Normalization has no effect on the boosting classifier.

We measure accuracy using the standard log-average miss rate for the detections [12], which is computed by averaging the miss rate at nine false positives per image (FPPI) rates evenly spaced between 10^{-2} to 10^{0}. A detection is labeled as correct if the area of overlap is greater than 50%.

We implement several baselines. The result of Dollár *et al*. [11], as reported in [12], is a 56% log-average miss rate using only static features and trained on the INRIA dataset [7]. Retraining on the Caltech training set reduced this error to 51%, which is close to the best reported results. By shrinking the model size from 128×64 to 64×32 and excluding occluded pedestrians from the training set we were able to reduce this rate to 45%. Likewise, using code from [16] we trained a HOG-SVM detector [7]. Again excluding occluded pedestrians, using a reduced model size, and shrinking the default HOG cell size to 4×4 pixels, we achieve 46% miss rate. *Our baselines slightly outperform the best reported results on the Caltech dataset.*

We now describe experiments testing each parameter. We perform our sweeps using boosting and the 10 static channel features described in Section 3. We explore each parameter sequentially while holding the others constant. For reference, we also always show the performance of our static detector. Lastly, we combine the optimal temporal features found for boosting with the static HOG features for use by linear SVMs. For the sweeps in Fig. 4 we used a slightly simplified evaluation criterion, resulting in minor differences from our final numbers reported in Fig. 5 (generated using the official evaluation code available from [12]).

Optical flow scale vs. frame skip: We first explore the space of two parameters; the scale of LK flows, σ, and the *skip* between two frames used to compute the temporal difference, m, see Fig. 4(a). For these experiments, we only use two frames, with the span n equal to m. We use D_0, where the first frame is the reference frame when computing differences. Observe that there exists a coherent relationship between miss rates and these two parameters. When the pair of frames are temporally nearby, stabilization plays a smaller role, since objects are relatively well aligned even without stabilization. As we increase the skip m between the pair of frames, stabilization becomes critical. We fix $\sigma = 16$ for all remaining experiments.

Ideally, the optical flow scale should roughly cover an object, and so would be defined relative to the size of the candidate window being evaluated. For simplicity, we implemented a fixed scale in our experiments, which still worked well because our datasets tend to contain objects at a single scale. Moreover, Fig. 4(a) shows stable performance over two octaves in scale space, indicating that precise scale selection may not be necessary in general.

Other forms of stabilization: In addition, we explored global 2D transformations for stabilizing videos including translation, similarity, and projective transformations. Our stabilization outperforms these considerably, see Fig. 4(b).

Multiframe: Given a fixed scale $\sigma = 16$, we now examine the question of the optimal multiframe span n, skip m, and reference frame. Certain combinations are not possible ($m > n$) and so cannot be evaluated. We find that a large span $n = 8$ and small skip value $m = 1$ performs best, although a larger skip $m = 4$ also does well, see Fig. 4(c). Given the reduction in computational complexity of $D(8, 4)$ over $D(8, 1)$, we fix $n = 8$ and $m = 4$. Using these settings, we find using the current frame, $\mathcal{I}_t$, as the reference achieves the best result, see Fig. 4(d). This yields the final multiframe motion feature of $D_0(n = 8, m = 4)$.

Rectification: We examine various strategies for feature rectification in Fig. 4(e), using three temporal differences across the LUV color channels. The "Max" scheme uses the maximum temporal difference across the 3 channels, while the "Lum" scheme just uses the luminance (L) channel. "Rect" refers to rectified features that are created by appending the absolute value of the positive and negative components of the difference feature $D_0(8, 4)$. "Abs" refers to simply taking the absolute value of the difference feature, while "Signed" refers to keeping the original signed feature. We see in Fig. 4(e) that the signed luminance feature outperforms all the other variants.

Normalization: We evaluate the impact of feature normalization in Fig. 4(f). The normalization has minimal effect on the performance of the boosting classifier, presumably because boosting classifiers can train more flexible decision boundaries that perform implicit normalization. However, explicit normalization appears vital for linear SVMs. Similar finding have been shown for static features such as HOG [7].

Previous work: In Fig. 5 we compare with previous work including 'MultiFtr+Motion' [28] (which uses motion features) and 'MultiresC' [23] (which uses static features trained on the same data as [12]). Our models considerably outperform prior work, achieving a five-fold reduction in false positives. Both boosting and SVM classifiers perform well, each being optimal for different ranges of FPPI. Fig. 6 shows several examples of detections using our approach compared to using static features alone. Several false detections are removed around the car's boundary as temporal features remove the ambiguities. Temporal features can also help discover missed detections, such as the pedestrian riding a bicycle in the second row.

4.2. Part detection

The MINDSEye video dataset [1] is a large collection containing hundreds of hours of video capturing everyday outdoor human interactions for military surveillance scenarios. It is one of the largest available datasets for multi-person pose estimation and multi-person action recognition (Fig. 7). Though scripted, it is a challenging testbed for video analysis. We have annotated human poses in a collection of 7 video clips with each 30-100 seconds in duration. The annotated frames are evenly split into training and testing, and used to evaluate the ability of our motion features to perform human pose estimation in video sequences.

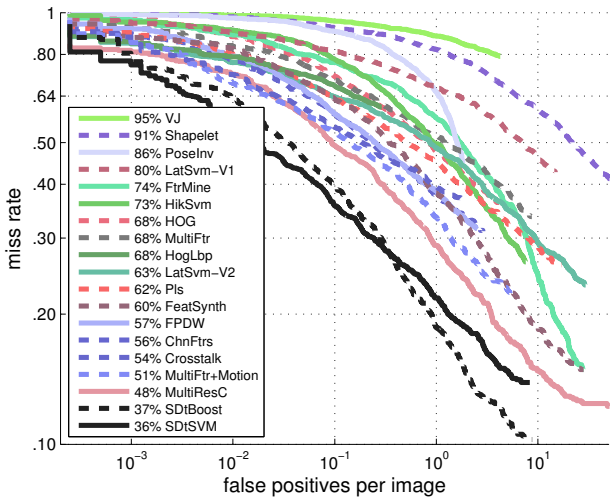

Figure 5: Comparison of log-average miss rate vs. False Positives Per Image (FPPI) between our approaches and previous methods on Caltech [12]. Our new temporal features lead to a significant improvement across all FPPI rates.

Baseline articulated part model: We describe our baseline articulated part model [31], and show how to extend it to incorporate our motion features. Let $l_i = (x_i, y_i)$ be the pixel location of part i. Given an image I, we score a collection of part locations $l = \{l_i\}$

$$\text{score}(I, l) = \sum_i w_i \cdot \phi(I, l_i) + w_s \cdot \text{spatial}(l) \quad (6)$$

where $\phi(I, l_i)$ is a HOG descriptor extracted from pixel location l_i in image I. The first term in (6) is an appearance model that computes the local score of placing filter w_i at location l_i using an inner-product. The second term is a shape prior that favors particular spatial arrangements of parts over others. From our perspective, we can be agnostic to its form so long as it is linearly parametrized and there exist tractable algorithms for computing the best scoring configuration $\max_l \text{score}_l(I, l)$. [31] describes efficient dynamic programming algorithms for inference, as well as efficient quadratic programming solvers for learning parameters $\{w_i, w_s\}$ given labeled training data.

Motion features: For our experiments, we simply augment the appearance descriptor to include both HOG and

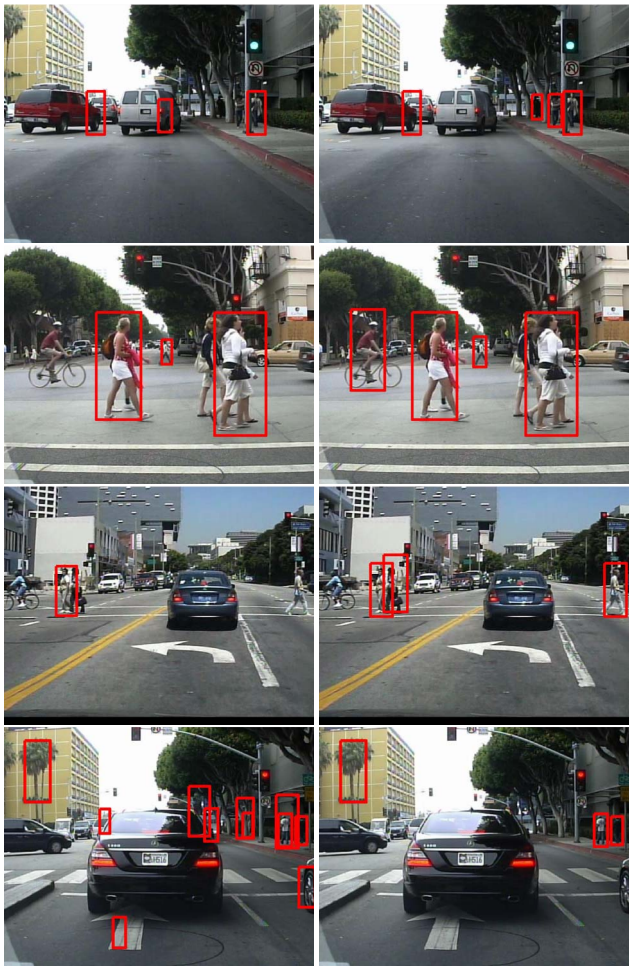

Figure 6: In the each row, we compare the results of two models; one trained only with static features (**left**), and the other trained with both static and our motion features (**right**). Note that our motion features help detect instances that are considered hard due to abnormal pose (biking) or occlusion, and significantly reduce false positives.

our motion feature:

$$\phi(I, l_i) = \begin{bmatrix} HOG[I, l_i] \\ D_0(8,4)[I, l_i] \end{bmatrix} \quad (7)$$

The above formulation allows us to easily incorporate our motion features into the existing pipeline at both test-time and train-time. Since the people in the MINDSEye dataset are significantly larger, we increased σ to 50.

Evaluation: We augmented the publicly-available code of [31] to use our motion features. We trained both a static-image pose detector and motion-augmented pose detector using the exact same training data, and present results in Fig. 7 and Table 1. For upper body parts, we see a large improvement in part localization accuracy (as measured by the fraction of times a predicted joint sufficiently overlaps the ground-truth). Overall accuracy across all joints increases from 57% to 60%, which is a reasonable improvement given the difficulty of the data. Multiple people often interact and occlude each other, making pose estimation and motion extraction difficult.

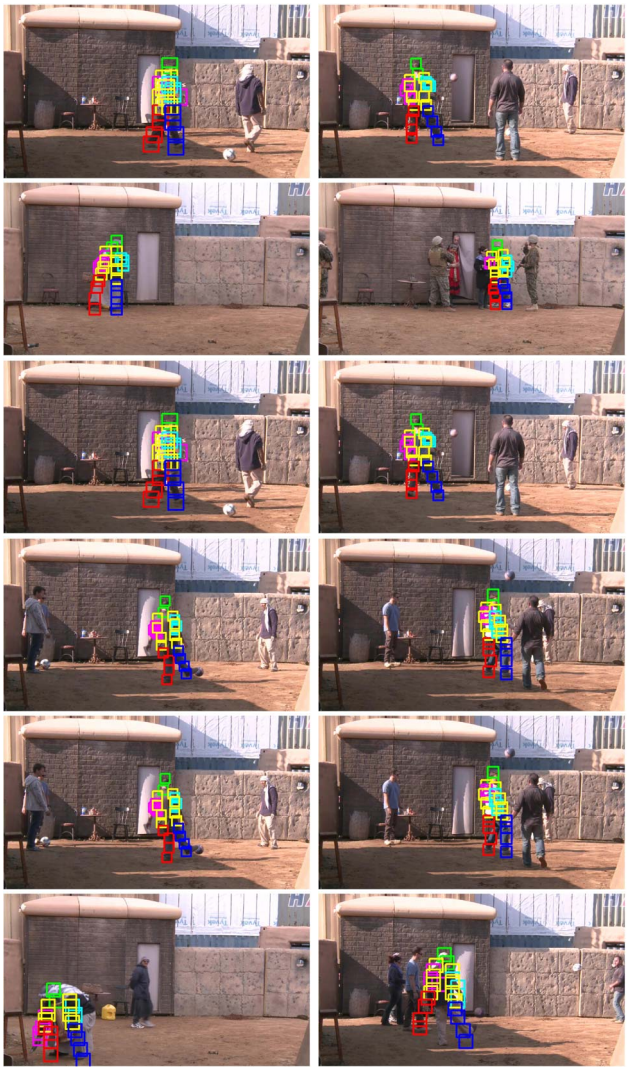

Figure 7: Pose estimation on MINDSEye test images. We show estimates from the pose model of [31] trained using our motion features. It outperforms static features, especially for instances with large motion, e.g. playing with a ball. The last row shows failure cases.

5. Conclusion

We described a family of temporal features utilizing weakly stabilized video frames. Weak stabilization enables our detectors to easily extract part-centric information by removing most camera- and object-centric motion. We experimentally show that simple temporal differences extracted across large time-spans are capable of producing

Features	HOG	HOG+Motion
Head	71.50%	**76.00%**
Upper arms	65.00%	**68.25%**
Lower arms	35.25%	**39.25%**
Upper legs	62.50%	**65.50%**
Lower legs	60.75%	**61.75%**
Overall	57.07%	**59.93%**

Table 1: Augmenting an articulated part model with our motion features produces consistently better part localizations. The gain from static features are not as dramatic as the result on Caltech, since other challenges, such as self-occlusion, inter-person occlusion, and a wider variety of poses, plays a role. Each body part (e.g., upper arm) contains 2 keypoints and is evaluated using standard criteria [31]. "Overall" refers to the average across all keypoints, making sure there is no double-counting.

state-of-the-art results on the challenging Caltech Pedestrian dataset. Finally, we show our features generalize to detecting individual body parts, as well as pedestrians.

Acknowledgments

Deva Ramanan was funded by NSF Grant 0954083 and ONR-MURI Grant N00014-10-1-0933. We thank Joseph Lim for helpful discussions.

References

[1] Minds eye dataset. http://www.visint.org/index.html. 2, 4, 6

[2] C. Anderson, P. Burt, and G. Van Der Wal. Change detection and tracking using pyramid transform techniques. In *Proc. SPIE Conference on Intelligent Robots and Computer Vision*, pages 300–305, 1985. 2

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, pages 1–8, 2008. 2

[4] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE TPAMI*, 33(3):500–513, 2011. 2

[5] R. Collins et al. *A system for video surveillance and monitoring*, volume 102. Carnegie Mellon University, the Robotics Institute, 2000. 2

[6] N. Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006. 2

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005. 1, 2, 3, 4, 5, 6

[8] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *ECCV*, pages 428–441, 2006. 1, 2

[9] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. *BMVC*, 2010. 2

[10] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCV VS-PETS*, 2005. 2

[11] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral channel features. *BMVC*, 2009. 2, 4, 5

[12] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE TPAMI*, 34(4):743–761, 2012. 2, 4, 5, 6

[13] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, 2003. 1, 2

[14] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008. 2

[15] A. Fathi and G. Mori. Human pose estimation using motion exemplars. In *ICCV*, pages 1–8, 2007. 2

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE TPAMI*, 2010. 2, 4, 5

[17] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE TPAMI*, 32(7):1239–1258, 2010. 2

[18] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ. Press, 2000. 1

[19] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *IEEE TPAMI*, 1997. 1

[20] M. Jones and D. Snow. Pedestrian detection using boosted features over many frames. In *ICPR*, 2008. 2

[21] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. *Spatial Coherence for Visual Motion Analysis*, pages 91–103, 2006. 2

[22] B. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 3

[23] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. *ECCV*, 2010. 6

[24] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics*, volume 4, pages 3099–3104. IEEE, 2004. 1, 2

[25] E. Shechtman and M. Irani. Space-time behavior based correlation. In *IEEE TPAMI*, 2007. 2

[26] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1

[27] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 2005. 1

[28] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR*, 2010. 2, 6

[29] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009. 1, 2

[30] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *IEEE TPAMI*, 2012. 2

[31] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392. IEEE, 2011. 2, 6, 7, 8

[32] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, volume 2, pages II–123. IEEE, 2001. 2