Segmentation of Dynamic Scenes with Distributions of Spatiotemporally Oriented Energies

Damien Teney
d.teney@bath.ac.uk
Matthew Brown
m.brown@bath.ac.uk

Media Technology Research Centre
Department of Computer Science
University of Bath
Bath, UK

Abstract

In video segmentation, disambiguating appearance cues by grouping similar motions or dynamics is potentially powerful, though non-trivial. Dynamic changes of appearance can occur from rigid or non-rigid motion, as well as complex dynamic textures. While the former are easily captured by optical flow, phenomena such as a dissipating cloud of smoke, or flickering reflections on water, do not satisfy the assumption of brightness constancy, or cannot be modelled with rigid displacements in the image. To tackle this problem, we propose a robust representation of image dynamics as histograms of motion energy (*HoME*) obtained from convolutions of the video with spatiotemporal filters. They capture a wide range of dynamics and handle problems previously studied separately (motion and dynamic texture segmentation). They thus offer a potential solution for a new class of problems that contain these effects in the same scene. Our representation of image dynamics is integrated in a graph-based segmentation framework and combined with colour histograms to represent the appearance of regions. In the case of translating and occluding segments, the proposed features additionally serve to characterize the motion of the boundary between pairs of segments, to identify the occluder and inferring a local depth ordering. The resulting segmentation method is completely model-free and unsupervised, and achieves state-of-the-art results on the SynthDB dataset for dynamic texture segmentation, on the MIT dataset for motion segmentation, and reasonable performance on the CMU dataset for occlusion boundaries.

1 Introduction

We are interested in the use of image motion and dynamics to aid the segmentation of videos, in addition to appearance cues such as colour and texture. For example, two adjacent regions of different colours, but moving together in the image, are likely to belong to a same object. Conversely, two objects may exhibit a similar texture but be moving differently. Though indistinguishable in individual frames, image dynamics then allow for separating them into different segments. Besides simple motion, another class of phenomena, called dynamic textures, are characterized by complex variations of appearance. Examples include a swirling cloud of smoke, reflections on water, or swaying vegetation in outdoor scenes. In order to

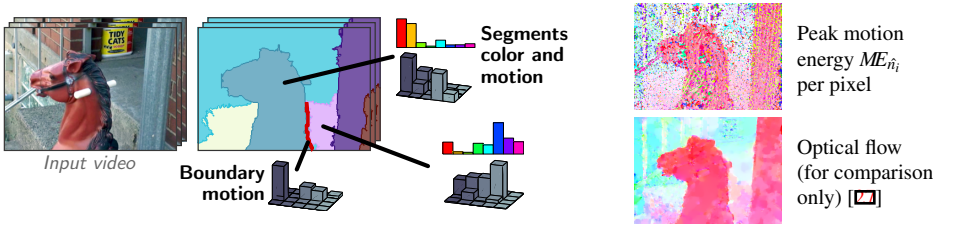


Figure 1: **(Left)** We represent dynamics in regions of the video with histograms of motion energies (HoME) measured at various space-time orientations. They are combined with colour histograms in a graph-based segmentation framework. Post segmentation, HoMEs are additionally used to compare the motion of boundaries with their adjacent segments'. We thereby identify the occluders and infer a local depth ordering. **(Right)** The peak motion energy $ME_{\hat{n}_i}$ at each pixel captures local variations of appearance; although comparable to a noisier version of optical flow, the full set of measurements, of higher dimension, offers a much richer representation of image dynamics (hue represents orientation, saturation represents velocity/magnitude.

reliably segment such patterns, image dynamics are even more crucial, though they cannot be easily captured with traditional methods based on optical flow.

In this paper, we present a method for extracting and representing image dynamics using 3D, spatiotemporal filters applied to the video volume. The response of these filters is turned into histograms of spatiotemporally oriented energies, accumulated across regions, and used within an existing video segmentation method [13]. Most interestingly, the same technique applies to dynamics arising from simple motions and complex dynamic textures alike. This strongly contrasts with existing work that focuses on either of these two classes of problems, or those based on optical flow and parametric motion models. Furthermore, in the case of rigidly moving objects, we use our representation of motion to assign each boundary to either of its two adjacent segments. The most similar is likely to be occluding the other, which allows us to infer a local depth ordering. The procedure proved very effective in practice, despite relying entirely on low-level image features, without assuming any particular motion model, with the advantage of operating completely unsupervised, *i.e.* without prior training.

In summary, our contributions consist of *(i)* a representation of image dynamics as histograms of spatiotemporally oriented energies, *(ii)* its application to the segmentation of videos, by adapting an existing method to identify segments of coherent dynamics *and* appearance, *(iii)* an additional, post-segmentation procedure to assign each resulting boundary to either of its adjacent segments, inferring a local depth ordering, and *(iv)* an extensive evaluation on a range of tasks, namely dynamic texture segmentation, motion layer estimation, and occlusion boundary detection. We demonstrate the wide applicability of the approach to problems previously studied separately, with results superior to a number of existing, task-specific methods.

2 Related work

Video segmentation Video segmentation has been actively studied; see *e.g.* [28] for a review. A number of methods extend image segmentation algorithms, using only colour and

texture to group perceptually homogeneous regions. Motion was also used as an additional feature (*e.g.* [5, 13] among many others) mostly with histograms of optical flow. In [13], the authors additionally use of the optical flow to define the connectivity (and thus possible groupings) between nearby voxels. In general, optical flow is limited by assumptions of *e.g.* brightness constancy or rigid motion, and is thus limited to image dynamics corresponding to actual displacements in the image. Patterns such as reflections on water or a dissipating cloud of smoke violate these assumptions. Moreover, despite modern advances, the extraction of optical flow remains a demanding process that may be unsatisfactory, conceptually and practically. In contrast, we use low-level image features obtained from simple filtering of the video, within a state-of-the-art segmentation framework [13].

Extraction of image dynamics Besides optical flow, image dynamics have been used more directly within appearance-based methods. Works on dynamic textures (such as the water or smoke examples mentioned above) used generative models, *e.g.* linear dynamical systems [5, 10]. Iterative fitting of such models with expectation-maximization was used to segment dynamic textures [2, 4], but proved computationally expensive, requiring manually defined initial segmentations. Good results were recently reported on dynamic texture segmentation [5, 14] using extensions of (static) texture descriptors, but with limited contributions to the study of dynamics. Decompositions of image dynamics in the frequency domain [6, 11], showed potential for separating motions occurring at different frequencies within a scene. Such "band-pass" decompositions are comparable to our use of spatiotemporal filters. Steerable filters [12] were proposed early as a way to extract optical flow [16], though the responses to a bank of such oriented filters actually provide much richer information than the optical flow. Indeed, two key advantages are, on the one hand, to allow capturing multiple oriented structures at any space-time location, and, on the other hand, to handle both motion (*e.g.* translating objects) and non-motion (*e.g.* flickering effects) dynamics within a unified framework. Derpanis *et al.* looked extensively into their use for recognition of dynamic textures and scenes [9], and inspired our work. An early attempt at grouping these features with mean-shift was proposed in [8] but with limited results.

Motion layers The segmentation of motion was examined mostly using optical flow and parametric (*e.g.* affine) motion models, either in an independent step (*e.g.* [1] among many others) or jointly with the extraction of optical flow [23, 24]. The end goal however remains the estimation of displacements in the image, which only correspond to a limited range of situations. A recurring challenge is in determining the optimal number of layers; [24] proved computationally very expensive for this reason. In comparison, our hierarchical segmentation produces several levels of segmentation, justified by different possible levels of interpretation of the scene.

Occlusion boundaries Identifying occlusion boundaries in videos was originally brought up by [18], noting that an occlusion boundary moves together with the occluding surface [18]. A number of works focused on the learning of classifiers to recognize this behaviour [17, 22], using static and flow-based features [15, 25, 26], and initial candidates from a static edge detector. Sundberg *et al.* [25] showed that motion compared between local neighbourhoods could separate layers and infer depth ordering. These results motivated our approach, which assigns boundaries to either of the adjacent segments. Interestingly, the candidate boundaries of [21, 22] result from the comparison of intensity histograms between halves of spatiotemporal (3D) patches. Although formulated very differently, this strongly resembles our histogram-based segmentation. In comparison to the above methods, we perform different steps (from candidate boundaries to global consistency) in a more unified manner.

While other methods report results on a single (middle) frame of short video sequences, we produce segmentations for entire videos. Finally, we do not rely on hard-coded parametric motion models or require any training.

3 Spatiotemporally oriented energies to capture dynamics

We first present a filter-based approach to extract, from a video, motion and variations of appearance. We will then show how histograms of such features integrate within an existing segmentation method, combined with static appearance cues (colour histograms).

Our approach is based on earlier work on steerable spatiotemporal filters [7, 12]. Similarly to 2D filters used to identify 2D structure (*e.g.* edges) in images, 3D filters can reveal structure in the video volume. Considering a Gaussian-like function of three variables $G(x,y,t) = e^{-(x^2+y^2+t^2)}$, we use the second order derivatives $G2_{\hat{\theta}}(x,y,t) = \frac{\partial^2 G}{\partial \hat{\theta}^2}$ and their Hilbert transforms $H2_{\hat{\theta}}(x,y,t)$, steered to a spatiotemporal orientation of unit vector $\hat{\theta}$ (the symmetry axis of the $G2$ filter). We denote the video volume of stacked frames $\mathcal{V}$, and the energy response for a given $\hat{\theta}$ is then measured by

$$E_{\hat{\theta}}(x,y,t) \;=\; (G2_{\hat{\theta}} * \mathcal{V})^2 + (H2_{\hat{\theta}} * \mathcal{V})^2 \quad , \tag{1}$$

where $*$ denotes the convolution. Note that the Hilbert transform corresponds to a phase shift of $\pi/2$, and the quadrature pair of filters $G2/H2$ allows for extracting spectral strength independent of the phase [12]. In the spatiotemporal frequency domain, a pattern moving in the video with a certain direction (*e.g.* rightwards) and velocity (*e.g.* 2 px/frame) corresponds to a plane passing through the origin [12]. Our representation of image dynamics is based on measurements of energy along a number of those planes. Parameterizing a plane by its unit normal $\hat{n}$ in the spatiotemporal frequency domain $(\omega_x, \omega_y, \omega_t)$, the motion energy ME along the plane is given by

$$ME_{\hat{n}}(x,y,t) \;=\; \sum_{i=0}^{N} E_{\hat{\theta}_i}(x,y,t) \quad , \tag{2}$$

with $N = 2$ the order of the derivative of the filter, and $\hat{\theta}_i$ filter orientations whose response lie in the plane $\hat{n}$ (see [8] for details). These motion energies are thus obtained by summing responses of filters consistent with the orientation of each plane. The motivation is to obtain a representation of dynamics only, and this effectively marginalizes the filter responses over appearance. Note that the resulting measurements $ME_{\hat{n}}$ can be compared to the extraction of optical flow, since they each correspond to a specific orientation and velocity (see Fig. 1, right). In comparison, the responses of individual filters (Eq. 1) only measured orthogonal motion with respect to the local gradient.

The above formulation suffers from two drawbacks. Firstly, due to the broad tuning of the $G2$ and $H2$ filters, energy responses (Eq. 1) arise in a range of orientations around their peak tunings. The effect propagates to the aggregated energy measurements (Eq. 2), whose values are heavily correlated across neighbouring planes (see Fig. 2). Secondly, the response to any particular filter depends on image contrast, and one cannot thus directly determine whether a high response is caused by a definite 3D structure matching the filter orientation or a faint match in a region of high contrast. We address these two issues by a non-linear scaling of $ME_{\hat{n}}$, first normalizing w.r.t. the strongest local energy measure, then as to emphasize the

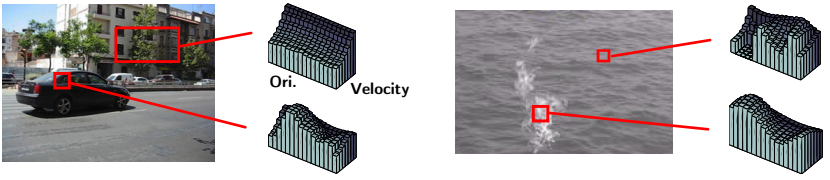


Figure 2: Actual HoMEs of real sequences (Eq. 2), visualized as 2D histograms, of image (spatial) orientations and (spatiotemporal) velocities (lighter colours represent higher velocities; a limited set of velocities is represented for compactness). **(Left)** The background is mostly static with a uniform range orientations, whereas the moving car produces a single mode in the histogram. **(Right)** The sea waves exhibit multiple motion modes; the upwards motion of the flame is more simply defined.

actual peak energies at each voxel:

$$ME'_{\hat{n}}(x,y,t) \;=\; ME_{\hat{n}}(x,y,t) \;/\; \max_{\hat{n}} ME_{\hat{n}}(x,y,t) \tag{3}$$

$$ME''_{\hat{n}}(x,y,t) \;=\; e^{\alpha(ME'_{\hat{n}}(x,y,t)-1)} \;. \tag{4}$$

In practice, we use a high parameter $\alpha = 1000$, typically. The feature vector of each pixel is finally the set of measurements $E''_{\hat{n}_i}$ for a number of vectors $\hat{n}_i$ (see Sect. 6), considered as a histogram and therefore normalized as to sum to 1.

4 Segmentation combining appearance and dynamics

We integrate our representation of image dynamics within the hierarchical, graph-based video segmentation method of [13], briefly reviewed below. The video volume is represented as a graph, where the nodes $\mathcal{N}_\ell = \{p_i\}_i$ initially correspond, at level $\ell = 0$, to all voxels of the video. The edges $\mathcal{E}_\ell$ between nodes initially correspond to a 26-connectivty, and are assigned a weight proportional to the similarity between nodes. To produce each level of segmentation, an agglomerative procedure iteratively removes edges from the graph, merging nodes into segments of larger and larger size, until some criterion is satisfied. The resulting graph at level ℓ is used as the starting point for the level $\ell+1$. Although the algorithm performs local decisions by considering, at each level, the edges in order of increasing weight, it ensures that segments can equally correspond to homogeneous (*e.g.* textureless) regions or to regions exhibiting coherent but large variations.

We denote the set of voxels of $\mathcal{V}$ assigned to a node (segment) p with $\mathcal{P}(p)$. The appearance of p is characterized by the histogram of colours H_p^{col} occuring within $\mathcal{P}(p)$ and by our histograms of motion energy H_p^{HoME}. As opposed to colour histograms, note that the HoME of single voxels (at level 0) already represent a distribution, with multiple entries. An edge between nodes p and q is assigned a low weight whenever colour and dynamics are both similar:

$$\text{weight}(p,q) = 1 - \Big(1 - d(H_p^{\text{col}}, H_q^{\text{col}})\Big)\Big(1 - d(H_p^{\text{HoME}}, H_q^{\text{HoME}})\Big) \;, \tag{5}$$

with $d(\cdot,\cdot)$ the Chi squared distance. One could introduce here a different weighting for colour and dynamics, as most authors do (*e.g.* [25]), and optimize results on a particular dataset. A generally optimal choice is however not obvious, and we chose to keep both features equally important.

5 Inferring occlusion boundaries and depth order

In scenes containing mostly rigid objects, once they have been segmented into different regions, we want to reason about their depth ordering. We use the heuristic that an occlusion boundary then moves together with the occluding surface [18]. This allows reasoning beyond the apparent adjacency of the segments in the video. Formally, a boundary between two segments p and q correspond to the voxels of the video

$$\mathcal{B}(p,q) \;=\; \text{dilate}_{3\times3}\big(\mathcal{P}(p)\big) \;\cap\; \text{dilate}_{3\times3}\big(\mathcal{P}(q)\big) \;, \tag{6}$$

i.e. the intersection of the voxel volumes of both nodes, each dilated with a 3×3 cube. This corresponds to a 2-voxel wide boundary wherever the nodes are adjacent in the video volume. We then characterize the motion of a boundary by accumulating HoMEs over its voxels. Therefore, the boundary motion, denoted H^{HoME}_{pq}, is easily compared with the motion within the adjacent segments, and the boundary is assigned to the most similar of the two:

$$\text{boundaryAssignment}(p,q) \;=\; \underset{p'=\{p,q\}}{\arg\min}\; d(H^{\text{HoME}}_{p'}, H^{\text{HoME}}_{pq}) \;. \tag{7}$$

This assignment is done on an individual basis, one pair of segments at a time, giving a *local* depth ordering of these segments (similarly as in [25]). Some global consistency is however ensured thanks to the prior segmentation, which operates on the whole video. Inferring such information at different levels of the segmentation hierarchy is justified by the different possible levels of interpretation of the scene.

Finally, in addition to the multi-level, hierarchical segmentation, we aggregate results of all levels into a single boundary map (Fig. 6). For any frame f, this map includes all boundaries between segments at the lowest level of the hierarchy (remember that the boundaries of higher levels are a subset of them) and assign, to each of them, the following strength:

$$\begin{aligned}\text{boundaryStrength}(p,q) \;=\; \text{maxLevel} + \frac{1}{\text{maxLevel}} \sum_{\ell=1}^{\text{maxLevel}} \text{weight}\big(\,\text{parent}_\ell(p),\; \text{parent}_\ell(q)\,\big) \\ \text{with}\quad \text{maxLevel} = \max\;\ell \quad \text{s.t.}\quad \mathcal{B}(p,q) \subset \mathcal{B}\big(\text{parent}_\ell(p), \text{parent}_\ell(q)\big) \;,\end{aligned} \tag{8}$$

where $\text{parent}_\ell(p)$ gives, for p, the node p' of a higher level such that $\mathcal{P}(p) \subset \mathcal{P}(p')$. The first term of Eq. 8 correspond to the highest level of the segmentation in which this boundary appears, and dominates the overall strength. The second term, comprised in $[0,1[$, is the average importance of the boundary over all levels of the hierarchy. It provides a finer estimate of the strength. For example, the boundary between the two segments remaining at the top level of the hierarchy will receive a high, but non-uniform strength.

6 Experimental evaluation

We evaluated our approach on a number of tasks: dynamic texture segmentation, motion segmentation, and identification of occlusion boundaries. These have previously been addressed with distinct task-specific methods, and no single dataset allows a comprehensive quantitative evaluation. We thus consider different benchmark datasets in turn. Remarkably, we obtained superior or competitive results on all datasets with a single method and identical parameters. We use $G2$ and $H2$ filters of scale $\sigma=1$ px. We measure motion energies along planes $\hat{n}_i$ corresponding to 16 spatial orientations and 10 speeds between 0 and 3 px/frame, in addition to flicker (infinite velocity). The resulting HoMEs have a dimension of 161. Colour histograms consist of 3 separate 10-bin histograms using Lab coordinates. As in [13], the

Method		K=2	K=3	K=4
No init., no training	**Proposed, colour + HoME, static segments**	93.6	89.7	88.5
	Proposed, colour + HoME, moving segments	86.3	79.5	74.4
	Proposed, colour only, moving segments	71.1	60.8	61.2
	GPCA [20]	54.8	55.4	54.9
With manual init.	LDT [8]	94.4	89.4	91.6
	DTM (IC) [8]	91.5	85.3	86.8
	DTM (CS) [8]	91.5	82.5	83.5
With training	$(\text{LBP/WLD})_{\text{TOP}}$ [8]	92.4	88.4	85.5

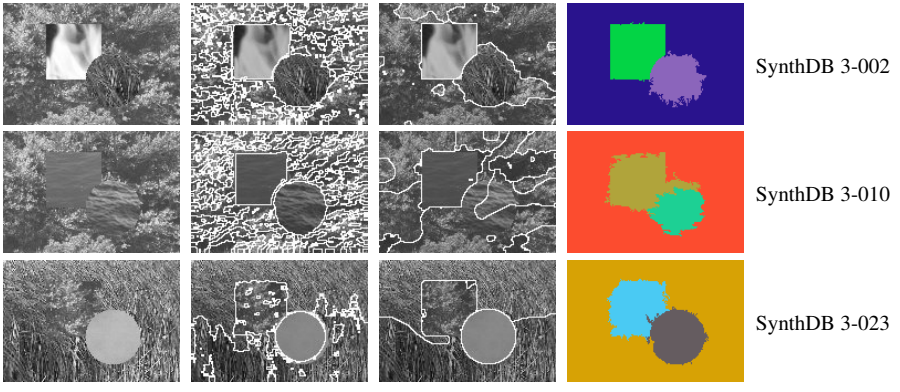
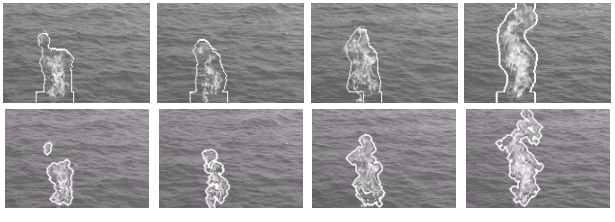


Figure 3: **(Above)** Segmentation of dynamic textures on the SynthDB dataset with K textures (Rand index in percent). **(Below)** Sample segmentations of textures of very similar appearance; image dynamics are crucial to distinguish them.



Chen *et al.* [8]
"Split-and-merge" algorithm causes square boundaries at the base of the flame.

Proposed, colour + HoME
Note the tighter and more precise boundaries.

Figure 4: Segmentation of the ocean-fire sequence (frames 4, 12, 27, 47).

segmentation at level $\ell=0$ is bootstrapped with edge weights set to Lab space distances between voxels (to avoid artefacts from histogram quantization) and segments of minimum area of 20 px (to ensure stable histograms). Input videos are filtered with a 2D Gaussian of size $\sigma = 0.5$ px to remove noise and compression artefacts. Our mixed Matlab/C implementation processed most of the videos of this evaluation in less than a minute on a standard laptop. The limitation is generally in the memory required to store the HoMEs of every voxel at level 0. We compare the use, as features, of colour histograms alone, in conjunction with HoMEs, and with histograms of optical flow (extracted with [27] and quantized in 2D histograms of 16 orientations and 10 magnitudes, similar to the HoMEs).

Method	Avg.	Car	Car2	Car3	Dog	Phone	Table	Toy	Hand	Person
Proposed, colour + HoME	**83.2**	90.0	64.5	79.6	95.9	56.1	93.7	90.8	94.5	83.4
Proposed, HoME only	82.2	88.4	63.7	83.0	95.7	57.0	90.9	87.0	87.5	86.8
Proposed, colour + flow [21]	79.9	81.1	53.3	89.8	93.5	66.5	89.5	87.1	62.9	95.6
Proposed, colour only	73.3	86.4	64.4	72.1	51.6	55.8	86.7	89.6	98.9	54.0
Layers++ [23]	77.5	61.2	51.2	77.8	96.4	56.7	90.9	83.2	81.4	98.6
nLayers [24]	82.3	83.6	58.9	76.6	97.4	57.8	97.9	85.8	88.1	94.4

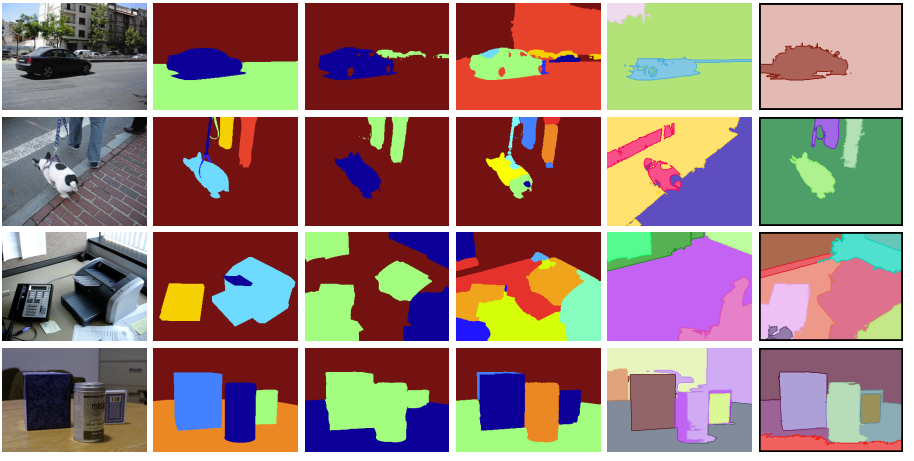


Figure 5: Motion segmentation on the MIT human-labeled dataset (Rand index in percent).

6.1 Dynamic texture segmentation

We segment dynamic textures with the SynthDB dataset [3], featuring composites of 2, 3, or 4 patches of real footage of fire, water, smoke, vegetation, etc. We generally perform better than existing methods (Fig. 3), which require a manual coarse initialization [2, 3] or a training stage [5] to learn an optimal distance function between features. We obtained best results by enforcing static segments (as in [2, 3]), forcing the accumulation of histograms over the frames of the video. We also show qualitative results on a classical sequence featuring fire over water (see Fig. 4); the complex moving boundaries of the flame are precisely estimated (video also provided as supplementary material).

6.2 Motion segmentation

We segment rigid motions with the MIT human-labeled dataset [23]. It features objects with intrinsic motion (*e.g.* car, dog) and parallax-induced motions at different depths. We correctly segment most objects with an impressive improvement over colour-only segmentation (see Fig. 5 and supplementary material). Quantitatively, we slightly surpass state-of-the-art methods [23, 24], which perform segmentation to help the extraction of motion (optical flow). Note that segmentation is our end goal, and we thus proceed the opposite way.

6.3 Occlusion boundaries

We detect **object boundaries** (Sect. 5, Eq. 8) on the CMU dataset [22]. Compared to motion segmentation, these boundaries do not have to define closed segments. Our assumption is, however, that our closed segments enforce some useful global consistency. We indeed obtain good performance, with unquestionable improvement over colour-based segmentations (see Fig. 6 and supplementary material). The segmentation is also improved over the "HoMEs only" segmentation, which indicates the importance of (static) appearance cues. Overall, we do not reach the performance of learning-based methods [15, 22, 25]. Since the baseline flow-based segmentation performs similarly, this does not point a limitation of the proposed features, but rather the suitability of learning-based methods on this dataset. The inevitable bias of human annotations favors a supervised training to combine static and motion cues. We observed however some failure cases due to the inability of HoMEs to capture motion in textureless regions. The above methods rely on optical flow and benefit from the usual regularization for this well-known aperture problem.

We finally infer the **depth ordering** of adjacent segments (Sect. 5, Eq. 7). The assignment for a boundary is made at the highest level it appears, and we measure agreement with ground truth pixel-wise over correct boundaries (as *e.g.* in [19]). We observed that the success of the assignment strongly depends on the quality of the segments identified in the first place. Correctly segmented scenes thus gave excellent results, and the overall performance (77%) is above the chance level of 50% (Fig. 6). We do not reach the state of the art [19, 25], which use parametric motion models, well suited to the rigid motions present in this dataset. A future direction could be the fitting of such models to our motion energies, though this may limit the generality of the current model-free approach.

7 Conclusions

We studied the use of image dynamics as a cue to help segment videos into coherent, physically meaningful regions. We extracted spatiotemporally oriented energies from the responses to a bank of second-order derivative of Gaussian filters, tuned to different (spatial) orientations and (spatiotemporal) speeds. Histograms of the resulting oriented energies proved effective as additional features to aid the segmentation of a wide variety of scenes, featuring simple motions and complex dynamic textures. We thus demonstrated that local, low-level image features describing image dynamics can provide powerful cues within a segmentation framework. An interesting avenue for future work is the study of other types of filters, *e.g.* higher-order derivatives of Gaussians, 3D Gabors, and Lognormal filters. These may offer better selectivity through finer tuning to spatiotemporal orientations. The use of multiple scales is another direction worth exploring. Finally, the proposed approach offers potential for handling scenes composed of mixtures of static, moving objects, and dynamic textures. An annotated dataset of such scenes would offer new challenges and may stimulate future advances in video segmentation.

Method		Boundary detection (F-measure, %)	Boundary detection (AP, %)	Depth ordering (%)
Unsupervised, model-free	Proposed, colour + HoME	60.7	56.3	77.0
	Proposed, HoME only	50.4	42.3	–
	Proposed, colour + flow	58.4	53.2	–
	Proposed, colour only	51.1	43.2	–
W/ supervised training	Stein and Hebert []	66.7	63.7	–
W/ parametric motion model	Palou *et al.* []	–	–	85.3

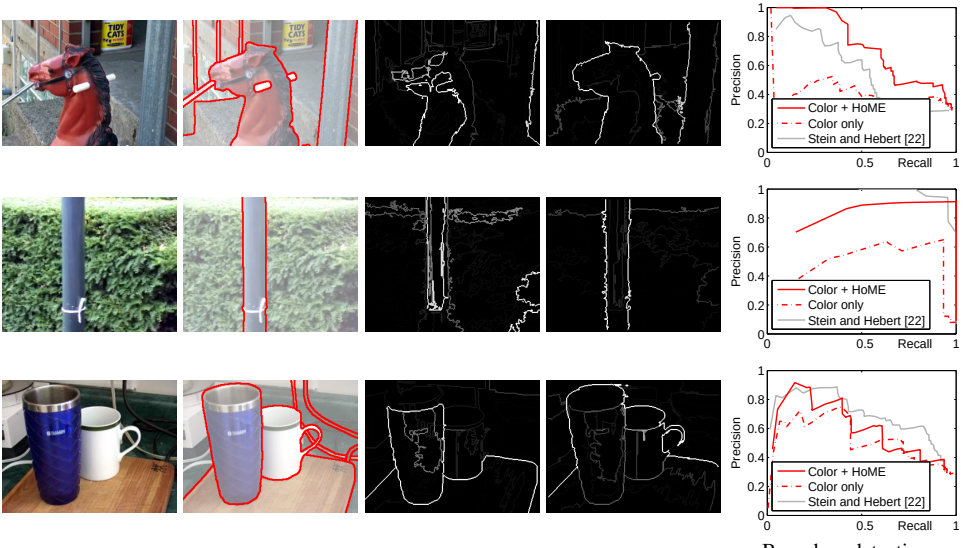


Figure 6: Detection of occlusion boundaries on the CMU dataset of sequences with camera translations. We segment different objects using their relative motions, caused by parallax at different depths. **(Bottom-right image)** The strength of an object boundary is not necessarily uniform. For example, the strong top boundary of the cups indicates a larger depth difference (with the background) than the bottom boundary (with the table).

References

[1] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. 2010.

[2] A. B. Chan and N. Vasconcelos. Layered dynamic textures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1862–1879, 2009.

[3] A.B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):909–926, 2008.

[4] A.B. Chan and N. Vasconcelos. Variational layered dynamic textures. In *CVPR*, pages 1062–1069, 2009.

[5] J. Chen, G. Zhao, M. Salo, E. Rahtu, and M. Pietikäinen. Automatic dynamic texture segmentation using local descriptors and optical flow. *IEEE Trans. Image Processing*, 22(1):326–339, 2013.

[6] D. Chetverikov and R. Peteri. A brief survey of dynamic texture description and recognition. In *Int. Conf. Computer Recognition Systems*, pages 17–26. Springer, 2005.

[7] K. G. Derpanis and J. M. Gryn. Three-dimensional nth derivative of gaussian separable steerable filters. In *ICIP (3)*, pages 553–456, 2005.

[8] K. G. Derpanis and R. P. Wildes. Early spatiotemporal grouping with a distributed oriented energy representation. In *CVPR*, pages 232–239, 2009.

[9] K. G. Derpanis and R. P. Wildes. Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(6):1193–1205, 2012.

[10] G. Doretto, P. Pundir, S.Soatto, and Y. N. Wu. Dynamic textures. In *IJCV*, pages 439–446, 2001.

[11] S. Dubois, R. Peteri, and M. Menard. Decomposition of dynamic textures using morphological component analysis. *IEEE Trans. Circuits and Systems for Video Technology*, 22(2):188–201, 2012.

[12] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, 1991.

[13] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[14] M. Haindl and S. Mike. Unsupervised dynamic textures segmentation. In *Computer Analysis of Images and Patterns*, volume 8047 of *LNCS*, pages 433–440. Springer Berlin Heidelberg, 2013.

[15] X. He and A. Yuille. Occlusion boundary detection using pseudo-depth. In *ECCV*, pages 539–552. 2010.

[16] D. J. Heeger. Model for the extraction of image flow. *J. Opt. Soc. Am. A*, 1987.

[17] A. Humayun, O. M. Aodha, and G. J. Brostow. Learning to find occlusion regions. In *CVPR*, pages 2161–2168, june 2011.

[18] K. M. Mutch and W. B. Thompson. Dynamic occlusion analysis in optical flow fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7(2):133–138, 1985.

[19] G. Palou and P. Salembier. Depth ordering on image sequences using motion occlusions. In *ICIP*, pages 1217–1220. IEEE, 2012.

[20] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[21] A. N. Stein and M. Hebert. Local detection of occlusion boundaries in video. In *In BMVC*, pages 407–416, 2006.

[22] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *IJCV*, pages 325–357, 2009.

[23] D. Sun, E. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *NIPS*, 2010.

[24] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, pages 1768–1775, 2012.

[25] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, pages 2233–2240, 2011.

[26] E. Turetken and A. Alatan. Temporally consistent layer depth ordering via pixel voting for pseudo 3d representation. In *3DTV Conference*, pages 1–4. IEEE, 2009.

[27] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-L1 optical flow. In *BMVC*, pages 1–11, 2009.

[28] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012.