Detecting salient motion by accumulating directionally-consistent flow

L. Wixson M. Hansen
Sarnoff Corporation
Princeton, NJ 08543
{lwixson,mhansen}@sarnoff.com

Abstract

Motion detection can play an important role in many vision tasks. Yet image motion can arise from "uninteresting" events as well as interesting ones. In this paper, salient motion is defined as motion that is likely to result from a typical surveillance target (e.g., a person or vehicle traveling with a sense of direction through a scene) as opposed to other distracting motions (e.g., the scintillation of specularities on water, the oscillation of vegetation in the wind). We propose an algorithm for detecting this salient motion that is based on intermediate-stage vision integration of optical flow. Empirical results are presented that illustrate the applicability of the proposed methods to real-world video. Unlike many motion detection schemes, no knowledge about expected object size or shape is necessary for rejecting the distracting motion.

1 Introduction

Motion detection can play an important role in many vision tasks, especially those related to detection and tracking for surveillance. Depending on the specific scene conditions, the difficulty of these tasks can vary widely. Some of the most challenging domains are those in which motion is being exhibited not just by the objects of interest, but also by other non-salient objects such as vegetation, shadows cast by vegetation, and specularities on water [3, 19].

Non-salient motions of this type are a common source of false positives for most simple motion-detection schemes, which either detect areas of frame-to-frame intensity change [1, 4, 15, 11, 24], or areas of intensity change with respect to some reference representation [9, 3, 23, 6].[1] When the reference representation is a learned probability distribution of intensities at each pixel, the system can, over time, learn not to report non-salient change, but it will still give rise to false positives until the reference representation has been learned [9]. Motion-based methods for change detection, such as the one presented in this paper, have the potential to be much more stable than those that rely on intensity representations.

Typical approaches for suppressing false positive detections are based on their aspect ratio, size, or magnitude of the frame-to-frame flow or normal flow [11, 6]. These approaches are not satisfying, since it is easy to construct counterexamples to such heuristics, such as the example we will present in Figure 3. For example, the frame-to-frame motion of the non-salient objects may be larger than that of the salient objects, especially if the non-salient objects are significantly closer to the camera or if the salient object is moving very slowly to avoid detection.

A more sound approach is to filter out false positives based on some aspect of the distance traveled by the object. Branches on a tree will stay roughly in the same place (or at least within some area) over time. The key problem is how to perform the tracking. Typically vegetation gives rise to many regions of change that are not constant in extent or motion from frame to frame, and which are therefore difficult to instantiate and track with a higher-level vision process. Some researchers have begun to examine ways of performing this detection using lower-level processing. For example, one approach uses multiple frames to construct "XT" or "YT"' spatiotemporal intensity slices from a sequence of frame-to-frame change images, and then to extract lines from these slices [13, 16] or even from the XYT spatiotemporal volume. An issue with this approach is how to select the image rows or columns to be used to construct the slice. For example, in scenes with extensive motion, it is not sufficient to simply project all the image columns onto a single X-

[1] These are by no means all the work in this area. A more thorough listing of past work can be found in [20].

0-7695-0164-8/99 $10.00 (c) 1999 IEEE

row in order to form the XT image. Another approach uses spatiotemporal filtering [19, 20]. However, this introduces an assumption that the object is moving with a certain velocity due to the velocity-dependent nature of the spatiotemporal filters.

In this paper, we take salient motion to be motion that tends to move in a consistent direction over time. We propose an approach that works by integrating frame-to-frame optical flow over time so that for each pixel it is possible to compute a rough estimate of the total image distance it has moved. On each frame, we update a salience measure that is directly related to the distance over which a point has traveled with a consistent direction. Because we use sub-pixel optical flow, the algorithm can track an object even if it is moving extremely slowly, and we can maintain our salience even if the object comes to a stop. (Of course, it may in some cases be desirable to suppress the salience of objects that stop for an extended time.) The algorithm is designed to minimize the salience of both easily-tracked oscillatory motion, such as a lone branch without leaves swaying periodically, as well as complicated assemblies of branches with fluttering leaves and occlusions. There are no user-controlled parameters relating to object size or intensity contrast; all parameters are related to velocity or distance traveled. Furthermore, the algorithm is not especially sensitive to these parameters; the same parameter settings are used for all the examples in this paper.

A related approach has recently been proposed in [17] to deal with detecting low-contrast moving objects in video from a moving airborne camera. Their approach, which uses normal flow to temporally propagate change energy, has been motivated by similar goals, but does not use consistency of direction as a filter.

2 Algorithm input

We shall denote an image at time t as either I_t or, when it is necessary to denote a specific image point $\mathbf{p}$, $I_t(\mathbf{p})$.

The computation of the salience measure takes as input a set of frame-to-frame optical flow fields. Let $\mathbf{F}(\mathbf{p}) = (F_x(\mathbf{p}), F_y(\mathbf{p}))$ denote an optical flow vector field that defines a 2D vector at each pixel location $\mathbf{p} = [x\ y]$. Such a flow field can be used to warp an image $I_t(\mathbf{p})$ to yield a new image. Let the function that performs such a warp be denoted as

$$warp(I_t, \mathbf{F}, \mathbf{p}) = I_t(\mathbf{p}')$$

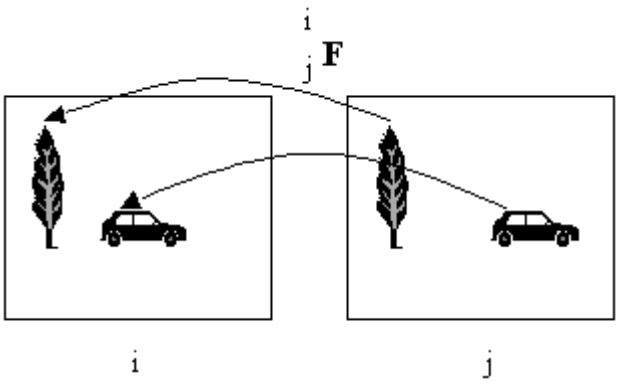


Figure 1. Illustration of notation used for flow fields. Flow field ${}^{i}_{j}\mathbf{F}$ maps coordinates in image j to image i. Flow field ${}^{j}_{i}\mathbf{F}$ maps coordinates in image i to image j.

where

$$\mathbf{p}' = \mathbf{p} + \mathbf{F}(\mathbf{p})$$

(It should be noted that when $\mathbf{F}$ has been computed to subpixel precision, then the x' and y' components of $\mathbf{p}'$ will not be integer values. Therefore $I_t(\mathbf{p}')$ must be computed using image interpolation [22]. We use bilinear interpolation in practice.) The result of applying the *warp* function at all pixel locations $\mathbf{p}$ shall be written as $warp(I, \mathbf{F})$.

The warp function also can be used to warp a 2D vector field $\mathbf{V}$. In this case, the warp function is applied to each component of the vector field individually:

$$warp(\mathbf{V}, \mathbf{F}, \mathbf{p}) = \begin{bmatrix} warp(V_x, \mathbf{F}, \mathbf{p}) \\ warp(V_y, \mathbf{F}, \mathbf{p}) \end{bmatrix}. \qquad (1)$$

Given two images I_i and I_j, the optical flow field that maps each pixel in I_j to a coordinate in I_i will be denoted as ${}^{i}_{j}\mathbf{F}$. This notation, developed by Craig [5], is illustrated in Figure 1. For images taken at two successive time instances t and $t+1$, the flow field ${}^{t}_{t+1}\mathbf{F}$ can be used to warp I_t into alignment with I_{t+1}, yielding a new image ${}^{t+1}I_t = warp(I_t, {}^{t}_{t+1}\mathbf{F})$.

In practice, we compute flow using a multi-resolution least squares technique [14, 2].[2] There are other variations [18] of this technique with better accuracy. However, since in general the motion in the scene will be complicated and non-rigid, it is unlikely that the

[2] In practice, there are well-known situations where the flow cannot be recovered if the image patch contains gradients in only one direction. Our algorithm handles this by computing only the normal flow in regions where only one image gradient is dominant.

specifics of the flow estimation will significantly impact the algorithm.

The difficulty of recovering perfect flow vectors is well-known [10]. In locations where there is occlusion, where the temporal sampling used for digitization is not fast enough to keep up with motion in the scene, or where there is insufficient texture, the computed flow vectors can be incorrect. We identify such flow vectors between two frames t and $t+1$ by performing forwards-backwards checking [8, 12] using the flow fields ${}^{t}_{t+1}\mathbf{F}$ and ${}^{t+1}_{t}\mathbf{F}$. The forwards-backwards checking examines whether the flow vectors in the two flow fields map to the same points. If not, the flow vector is set to 0. More specifically, ${}^{t}_{t+1}\mathbf{F}(\mathbf{p})$ is reset to [0 0] if $\|{}^{t}_{t+1}\mathbf{F}(\mathbf{p}) + {}^{t+1}_{t}\mathbf{F}(\mathbf{p} + {}^{t}_{t+1}\mathbf{F}(\mathbf{p}))\| > k_c$. I.e., the two flow vectors should cancel each other. The constant k_c is the pixel distance by which the two flow vectors can differ. Generally, when flow vectors are incorrect this distance will be large, so in practice $k_c = 3$ produces adequate checking.

3 Cumulative flow and salience

Theoretically, given perfect frame-to-frame flow fields and perfect image warping, one could track an image point from I_i to I_j by using the frame-to-frame flow fields ${}^{t}_{t+1}\mathbf{F}$ for $t = i, \ldots, j-1$. More specifically, the frame-to-frame flow fields could theoretically be combined into a "cumulative" flow field ${}^{i}_{j}\mathbf{C}$, as shown in Figure 2. This can be defined as

$$ {}^{i}_{j}\mathbf{C} = \begin{cases} {}^{i}_{j}\mathbf{F} & \text{if } j = i+1 \\ \\ \mathbf{\Delta}_j + warp({}^{i}_{j-1}\mathbf{C},\, {}^{j-1}_{j}\mathbf{F}) & \text{if } j > i+1 \end{cases} \tag{2} $$

where $\mathbf{\Delta}_j$ is the contribution, to the cumulative flow, of the frame-to-frame flow from frame $j-1$ to frame j. Theoretically, $\mathbf{\Delta}_j$ is simply ${}^{j-1}_{j}\mathbf{F}$, but this will change for the measures we develop below.

The cumulative flow field defined above can be used to measure the distance between each image point's location in a reference image I_i and its location in a subsequent image I_j. We will now develop a variation on this measure that accumulates the distance that each point travels in a consistent x- or y-direction, and relate this to a measure of salience. This measure will also be a vector field over the image, and will be denoted $\mathbf{S}_j$.

Our desired cumulative measure must have two properties. First, it must take on values that, for each point, are proportional to the distance that point has traveled in a consistent x- or y- direction. Second, since a flow field is rarely perfect, and since a salient object may temporarily pass behind small occlusions,

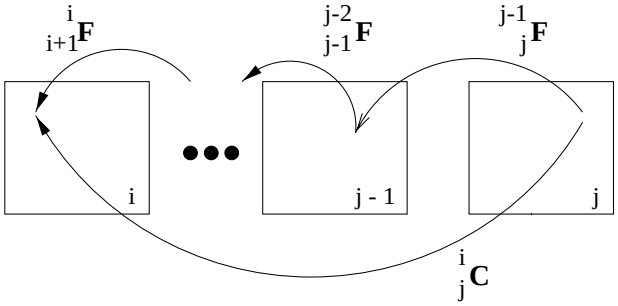


Figure 2. Given perfect frame-to-frame flow fields F, the theoretical cumulative flow field C would be identical to that obtained by composing the individual frame-to-frame flow vectors.

we would like the accumulation to be tolerant of small temporal gaps in the frame-to-frame tracking of a point where the frame-to-frame flow is incorrect.

3.1 The salience field

We now define a vector-valued salience measure $\mathbf{S}_j$ with the first property, i.e. it takes on values that, for each point, are related to the distance that point has traveled in a consistent x- or y- direction. This measure is similar to the theoretical cumulative flow field, except that we augment the system with a method of resetting the salience to 0 when the direction of each tracked point's flow reverses course, and use an "extended" flow field ${}^{j-1}_{j}\mathbf{E}$ rather than the original flow field ${}^{j-1}_{j}\mathbf{F}$ for each new frame j. The extended flow field is introduced to handle errors and occlusions that occur in real flow fields and will be explained in the next section; for the time being it suffices to consider it identical to the original flow field.

Given a new frame j, the computation of the salience measure is divided into three steps. The first simply updates an intermediate measure $\mathbf{S'}_j$ in the same manner as the theoretical cumulative flow updating shown in Equation 2, using the extended flow field:

$$ \mathbf{S'}_j := \begin{cases} \mathbf{0} & \text{if } j = 0 \\ \\ \mathbf{\Delta}_j + warp(\mathbf{S}_{j-1},\, {}^{j-1}_{j}\mathbf{E}) & \text{otherwise} \end{cases} \tag{3} $$

The second and third steps detect locations that have reversed direction, and reset their salience to zero. Detecting direction reversals is non-trivial, as it is common for a point's flow to reverse course slightly on some frames either due to errors in flow computation or occasional small backwards movement. Therefore,

to detect reversals in course we maintain a "maximum salience" 2D vector field that holds for each point the maximum value of the x- and y- components that the point's salience has taken on since the salience at that point was last reset. Direction reversals are detected when the maximum salience of a point is above some threshold k_s but the point's current salience is below some fraction k_r of the maximum.

We reset the salience separately for motion in the x- and y-directions, so that the overall salience magnitude at a point is not reset to 0 if the point reverses course in one direction but not the other (for example a person zigzagging while running forward).[3]

Let us now turn to the specifics of the second and third steps. In step two, the maximum cumulative flow field $\mathbf{M}_j$ is computed by warping it from the previous frame and updating those locations at which the one component of the salience vector is directionally consistent with the maximum cumulative flow vector and has a larger magnitude than the corresponding component of the maximum cumulative flow vector. Specifically, the x-component of the maximum cumulative flow vector on frame j, $M_{j,x}$, is updated at each point $\mathbf{p}$ as follows. Let m_x be the value of the x-component of the maximum cumulative flow vector at point $\mathbf{p}$'s location in the previous frame $j-1$, i.e. $m_x = \mathbf{M}_{j-1,x}(\mathbf{p} + {}^{j-1}_{j}E(\mathbf{p}))$. Then

$$M_{j,x}(\mathbf{p}) := \begin{cases} S'_{j,x}(\mathbf{p}) & \text{if } \mathrm{sign}(S'_{j,x}(\mathbf{p})) = \mathrm{sign}(m_x) \text{ and} \\ & \mid S'_{j,x}(\mathbf{p}) \mid > \mid m_x \mid \\ m_x & \text{otherwise} \end{cases} \tag{4}$$

The y-component, $M_{j,y}$, is updated similarly.

Finally, the third step detects direction reversals and resets the appropriate x- or y- component of the salience measure accordingly. The x-component of the salience measure, $S_{j,x}$, is assigned as follows.

$$S_{j,x}(\mathbf{p}) := \begin{cases} 0 & \text{if } \mid M_{j,x}(\mathbf{p}) \mid > k_s \text{ and} \\ & \mid S'_{j,x}(\mathbf{p}) - M_{j,x}(\mathbf{p}) \mid / \mid M_{j,x}(\mathbf{p}) \mid > k_r \\ S'_{j,x}(\mathbf{p}) & \text{otherwise} \end{cases} \tag{5}$$

If $S_{j,x}(\mathbf{p})$ is reset to 0, $M_{j,x}(\mathbf{p})$ is also reset to 0.

The y-component of the salience measure, $S_{j,y}$, is computed similarly. Typically the minimum salience k_s is set to 8 to ensure that some minimal salience has a chance to accumulate before it can be reset to 0. The fractional change k_r is typically set to .1, indicating that if the cumulative flow drops to 90% of the largest value previously observed, a direction change is occurring. The precise setting is not particularly important, since in general pixels on vegetation will exhibit direction reversals that represent large percentage changes relative to their maximum value.

[3] This can in some situations mean that zigzagging movement while running along the image diagonal may have slightly different resetting properties than those moving along image axes, but we have not observed any difficulties to date.

3.2 The extended flow field

To achieve robustness to errors in computed flow and temporal gaps created when a moving object temporarily passes behind small occlusions, we update the salience measure using an "extended" flow field ${}^{j-1}_{j}\mathbf{E}$ rather than the original flow field ${}^{j-1}_{j}\mathbf{F}$ for a new frame j. The extended field is derived from the original by checking for each point $\mathbf{p}$ in the original flow field, whether there exists a scalar multiple s of the original vector ${}^{j-1}_{j}\mathbf{F}(\mathbf{p})$ that extends the vector so that it connects to a location with large salience. More precisely, suppose we have $\mathbf{S}_{j-1}$, the salience measure from the previous frame. Then the vector-valued salience measure $\mathbf{g}$ at point $\mathbf{p}$'s location in the previous frame, assuming an extension by a factor of s, is $\mathbf{g} = \mathbf{S}_{j-1}(\mathbf{p} + s\,{}^{j-1}_{j}\mathbf{F}(\mathbf{p}))$. We test whether there exists an $s > 1$ that meets the following five criteria:

1. The flow vector to be extended must be large enough to be significant. Specifically, $\| \; {}^{j-1}_{j}\mathbf{F}(\mathbf{p}) \; \| \geq k_f$, where k_f is a user-specified distance (typically 1).

2. The extended flow vector can't be more than k_e pixels longer than the original vector. Specifically, $\| \; s\,{}^{j-1}_{j}\mathbf{F}(\mathbf{p}) - {}^{j-1}_{j}\mathbf{F}(\mathbf{p}) \; \| < k_e$ where k_e is a user-specified distance (typically 6).

3. The point to which the flow is to be extended must have a reasonably large salience. Specifically, $\| \; \mathbf{g} \; \| \geq k_g$, where k_g is a user-specified salience (typically 15).

4. The salience magnitude resulting from the extension must be more than the salience that would be obtained without an extension. Specifically, $\| \; \mathbf{g} \; \| > \| \; \mathbf{S}_{j-1}(\mathbf{p} + {}^{j-1}_{j}\mathbf{F}(\mathbf{p})) + {}^{j-1}_{j}\mathbf{F}(\mathbf{p}) \; \|$.

5. The vectors ${}^{j-1}_{j}\mathbf{F}(\mathbf{p})$ and $\mathbf{g}$ must lie in the same quadrant (i.e., the signs of their components must be identical).

If all of the above criteria are met, then we select the s that maximizes $\| \; \mathbf{g} \; \|$ and assign:

$$\begin{aligned} {}^{j-1}_{j}\mathbf{E}(\mathbf{p}) &:= s\,{}^{j-1}_{j}\mathbf{F}(\mathbf{p}) \\ \mathbf{\Delta}_j(\mathbf{p}) &:= \mathbf{0} \end{aligned}$$

This has the effect of setting the flow vector to be the extended flow vector, but the salience update term to 0. Intuitively, this allows the salience value of the tracked point to remain the same as that of the point to which it has been linked by the extension, but not to increase. The motivation for this policy is that since the flow was not actually observed, it should not increment the salience.

If not all of the criteria for extending the flow vector are met, then the extended flow and salience update is identical to the original flow:

$$\begin{aligned} {}^{j-1}_{j}\mathbf{E}(\mathbf{p}) &:= {}^{j-1}_{j}\mathbf{F}(\mathbf{p}) \\ \mathbf{\Delta}_j(\mathbf{p}) &:= {}^{j-1}_{j}\mathbf{F}(\mathbf{p}) \end{aligned}$$

4 Empirical studies

Figure 3 illustrates the algorithm on a challenging video sequence in which camouflaged soldiers are visible as very small objects while bushes in the foreground are large and sway wildly. To the human eye, the people are not visible in still frames from the sequence, and can only be seen when the sequence is played as a movie. Column 2 of the figure shows the x-component of the frame-to-frame flow. The regions corresponding to the salient objects (people at a distance) has been circled. The frame-to-frame velocity of the people varies from .6 to 2.5 pixels/frame, while that of the vegetation varies from 0 to 12 pixels/frame. Clearly the people cannot be distinguished from the foreground clutter on the basis of their size or their frame-to-frame motion magnitude. Column 3 of the figure shows the evolution of the x-component of the salience measure, $S_{j,x}$. Over time, $S_{j,x}$ for pixels on the soldiers increases (on the rightwards-moving soldier) or decreases (on the leftwards-moving soldier).

Notice that salient objects leave a streak behind them in the salience imagery. This is because the salience of a pixel location persists indefinitely until it is reset by a direction reversal. This policy has the benefit that it allows the salience measure to be largely unaffected by variations in the object velocity, even if the object comes to a stop. The trail could even be useful for further analysis or display of the object's history. On the other hand, in applications where objects paths cross or where an accurate delineation of the object is desired, further techniques can be applied to cause the trail to decay where it does not lie on the salient object. This will be discussed further below.

The magnitude of the salience is shown in Column 4. Over all the frames in the sequence, the salience magnitude found on the vegetation is no more than 55. In the first frame shown, (frame 37), the salience magnitude of the object is 31, so it would not yet be distinguishable from vegetation by thresholding. But by the second frame (frame 70), its salience magnitude is 60. Until this point in the sequence, the salience has increased slowly compared to the actual distance traveled by the object (160 pixels). This is because the object is so small that it is difficult to extract reliable flow vectors and so the flow vector extension is being used heavily, which does not increase salience. After frame 70, however, the object increases slightly in size and flow can be more reliably computed, so salience increases directly in proportion to the distance traveled. By the time the object reaches its leftmost position in the final frame (frame 150), its salience is 140. A second object also becomes visible, moving leftwards, in the third frame (frame 113). Its salience increases rapidly, since its flow is reliable. (The small linear extension protruding ahead of the object is the person's shadow on the ground.)

Figure 4 provides more examples of the algorithm on three other sequences. *Identical algorithm parameters were used for all four sequences.* In the top example, a person walks right to left while a fan blows the leaves of a potted plant at the left side of the image. The largest computed salience magnitude on the leaves was 25, while that of the person quickly rises with the distance traveled. In the frame shown, the person has traveled approximately 150 pixels and the typical salience of a pixel on the person is 140.

In the middle example a person walks upper-left to lower-right against a background of gently-swaying tree branches. The largest computed salience magnitude on the branches was 14, while that of the person rises quickly. In the frame shown, the person has traveled 126 pixels and his salience is approximately 122.

In the bottom example a person walks top to bottom while the branches on the tree sway violently in a strong wind. Furthermore, a car is visible for a brief period in the upper left corner as it moves from behind the tree and off the top edge of the frame. The largest computed salience magnitude on the tree was 35. Again, the salience of the person and vehicle rises quickly. In the frame shown, the person has traveled 48 pixels and his salience is approximately 45. His salience increases further as he travels further in subsequent frames.

5 Temporal decay

As noted above, salient objects leave a streak behind them in the salience image. In many applications, it may be desirable to add a mechanism that allows the salience to either decay gradually over time or rapidly

be set to 0 once the object has moved past. For example, this might be desirable if one wished to use the salience magnitude to delineate the object. The appropriate approach depends on the application. Here we report one possible mechanism, whose goal is to reset the salience of a pixel to 0 when the moving object no longer is imaged in the pixel.

We achieve this goal by determining, for each pixel $\mathbf{p}$, whether there exists another pixel $\mathbf{p}'$ within some distance k_d whose frame-to-frame flow magnitude exceeds that of $\mathbf{p}$ by more than some factor k_a. If so, then $\mathbf{S}_j(\mathbf{p})$ and $\mathbf{M}_j(\mathbf{p})$ are reset to $\mathbf{0}$ before the next new frame is processed. The intuition behind this scheme is that if there is nearby motion that is substantially larger than the motion at this pixel, then this this is likely to have happened because the object has moved off this image pixel.

This approach usually gives good results (see the bottom two examples in Figure 4). However, there are some scenarios where it does not suffice. Consider, for example, an intruder crawling slowly beneath waving tree limbs. The proximity to the tree limbs might result in the suppression of the intruder's salience. Obviously, there exist a gamut of variations that might be appropriate, such as basing the reset on whether the salience at $\mathbf{p}$ changes by some amount within a user-specified time window.

6 Discussion

This paper has outlined a salience measure that at each pixel is based on the straight-line distance that the pixel has moved in a consistent direction. Our examples have shown that objects moving in a straight line rapidly take on salience magnitudes that are significantly larger than that of vegetation. This suggests that for surveillance tasks, it might be possible to trigger a detection alarm at a pixel when the magnitude of its salience exceeds a threshold, and that it will be possible to choose a threshold that results in significantly fewer false positives than more conventional change detection schemes. This threshold would be based on the expected amount of side-to-side movement of vegetation in the scene. Alternatively, other more sophisticated analysis techniques might be applied to the salience "trails" left by objects.

The algorithm has some further advantages. It does not need to explictly detect and track hypothetical targets to assess their salience. It does not make assumptions about the size or intensity contrast of salient objects. Because it uses multi-resolution optical flow it is applicable to a broad range of image velocities, and can even handle image stops. Of course, it still is possible for salient objects to move either so slowly or so quickly that the flow is not reliable. To handle very slow-moving objects, it may be necessary to select among various temporal scales when computing flow. However, in surveillance scenarios involving objects that move by only a small fraction of a pixel per frame, shape change as recovered from stereo [7] is a more appropriate cue than motion.

The algorithm also has some weaknesses. An object that moves in a straight line but oscillates forwards and backwards, such as taking two steps forward and then one backward would have low salience. Again, in surveillance scenarios where subjects are actively trying to fool the salience measure, it is probably necessary to supplement this motion-based method with a shape-based method such as stereo. Another issue is computational expense. However, this issue is only temporary; as flow-warping hardware becomes widely available, the salience computation will rapidly become tractable.

Finally, optical flow has received widespread criticism as being inaccurate and error-prone. However, our results show that it can nonetheless be used to define effective salience measures. Future work, therefore, may examine its use in other grouping measures, such as those of Williams [21].

References

[1] C.H. Anderson, P.J. Burt, and G.S. van der Wal. Change detection and tracking using pyramid transform techniques. In *SPIE Vol. 579 - Intelligent Robots and Computer Vision*, pages 72–78, 1985.

[2] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, 1992.

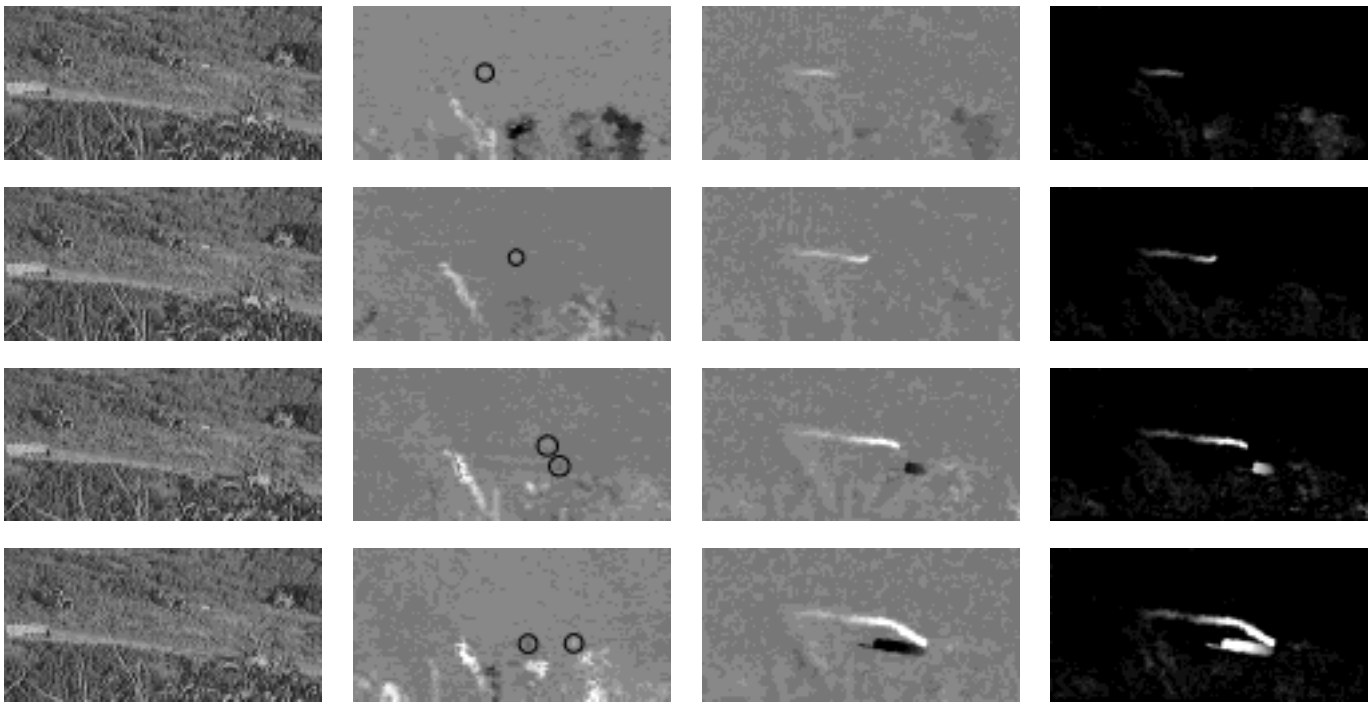
[3] T. Boult, R. Michaels, A. Kerkan, P. Lewis, C. Powers, C. Qian, and W. Yin. Frame-rate multi-body tracking for surveillance. In *Proceedings of the DARPA Image Understanding Workshop*, 1998.

[4] P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera: An application of dynamic motion analysis. In *Proceedings of the IEEE Workshop on Motion*, March 1989.

[5] J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.

[6] R. Cutler and L. Davis. View-based detection and analysis of periodic motion. In *International Conference on Pattern Recognition*, 1998.

[7] C. Eveland and K. Konolige. Background modeling for segmentation of video-rate stereo sequences. In *IEEE*



Raw image I_j | x-component of frame-to-frame flow, ${}^{j-1}_{j}F_x$ | x-component of cumulative consistent flow, $S_{j,x}$ | Magnitude of salience $\| \mathbf{S}_j \|$

Figure 3. Illustrative example of salience detection on four frames from a challenging video sequence, proceeding temporally downward. The center two columns contain signed imagery, so medium gray represents 0. Column 2 contains the x-component of the frame-to-frame flow (after zeroing of those flow vectors that fail the forwards-backwards check). The frame-to-frame flow corresponding to the salient objects (people at a distance) has been circled. Clearly the people cannot be distinguished from the foreground clutter on the basis of their size or their frame-to-frame flow magnitude. See text for details.

Conference on Computer Vision and Pattern Recognition, 1998.

[8] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6 (1), 1993.

[9] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[10] G. Halevi and D. Weinshall. Motion of disturbances: Detection and tracking of multi-body non-rigid motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[11] A. Lipton, H. Fujiyoshi, and R. Patil. Moving target classification and tracking from real-time video. In *Workshop on Applications of Computer Vision*, 1998.

[12] James J. Little and Walter E. Gillett. Direct evidence for occlusion in stereo and motion. In *Proceedings of the European Conference on Computer Vision*, 1990.

[13] F. Liu and R. Picard. Finding periodicity in space and time. In *Proceedings of the International Conference on Computer Vision*, 1998.

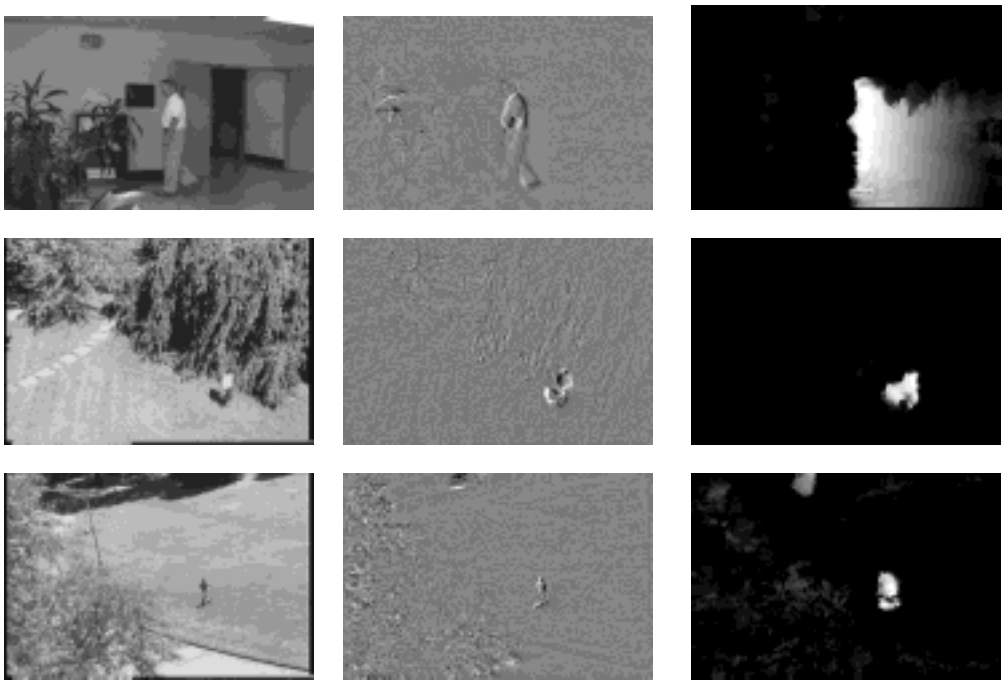
[14] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo. In *DARPA Image Understanding Workshop*, 1981.

[15] F. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP : Image Understanding*, 60(2), September 1994.

[16] S. Niyogi and E. Adelson. Analyzing and recognizing walking figures in xyt. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[17] R Pless, T. Brodsky, and Y. Aloimonos. Independent motion: The importance of history. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[18] Joseph Weber and Jitendra Malik. Robust computation of optical flow in a multi-scale differential frame-



Example raw image. Frame-to-frame difference. Salience magnitude, $\| \mathbf{S}_j \|$

Figure 4. Salience measures from three other video sequences. The bottom two rows contain the salience magnitude after supressing the salience trails, as described in Section 5.

work. In *Proceedings of the International Conference on Computer Vision*, 1993.

[19] R. Wildes. A measure of motion salience for surveillance applications. In *IEEE International Conference on Image Processing*, 1998.

[20] R. Wildes and L. Wixson. Detecting salient motion using spatiotemporal filters and optical flow. In *Proceedings of the DARPA Image Understanding Workshop*, 1998.

[21] L. Williams and K. Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. In *Proceedings of the European Conference on Computer Vision*, 1998.

[22] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1992.

[23] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, July 1997.

[24] S. Yalamanchili, W.N. Martin, and J.K. Aggarwal. Extraction of moving object descriptions via differencing. *Computer Graphics and Image Processing*, 18:188–201, February 1982.