Detecting Salient Motion by Accumulating Directionally-Consistent Flow

L. Wixson, *Member*, *IEEE Computer Society*

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.

Index Terms—Motion detection, optical flow, vegetation, motion salience.

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 nonsalient objects such as vegetation, shadows cast by vegetation, and specularities on water [3], [29].

Nonsalient 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], [13], [17], [22], [18], [26], [34], or areas of intensity change with respect to some reference representation [8], [25], [16], [11], [3], [33], [7]. When the reference representation is a learned probability distribution of intensities at each pixel, the system can, over time, learn not to report nonsalient change but it will still give rise to false positives until the reference representation has been learned [11]. 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 [18], [7]. These approaches are not satisfying, since it is easy to construct counterexamples to such heuristics, such as the example we will present in Figs. 3 and 4. For example, the frame-to-frame motion of the nonsalient objects may be larger than that of the salient objects, especially if the nonsalient 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 [20], [23] 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-row in order to form the XT image. Another approach uses spatiotemporal filtering [29], [30]. 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 subpixel 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 [24] 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. Directional consistency has also been used in algorithms for robustly detecting locally defined moving edges [15], [14]. Neither of these related approaches has been applied to imagery containing moving vegetation. Due to the challenging nature of the image motions in such sequences, as will be described, more intricate methods are required.

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

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

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 [32]. 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 author is with ParentWatch, Inc., 49 West 37th Street, New York, NY 10018. E-mail: lwixson@wixson.com.*

Manuscript received 28 Mar. 2000; accepted 21 Apr. 2000.
Recommended for acceptance by R. Collins.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 109665.

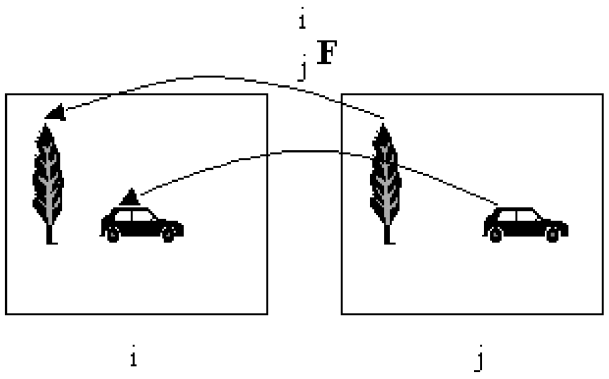

Fig. 1. Illustration of notation used for flow fields. Flow field ${}^i_j\mathbf{F}$ maps coordinates in image j to image i. Flow field ${}^j_i\mathbf{F}$ maps coordinates in image i to image j.

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:

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

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 [6], is illustrated in Fig. 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 multiresolution least squares technique [21], [2].[1] There are other variations [28] of this technique with better accuracy. However, since in general the motion in the scene will be complicated and nonrigid, it is unlikely that the specifics of the flow estimation will significantly impact the algorithm.

The difficulty of recovering perfect flow vectors is well-known [12]. 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 [10], [19] 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 Fig. 2. This can be defined as

1. 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.

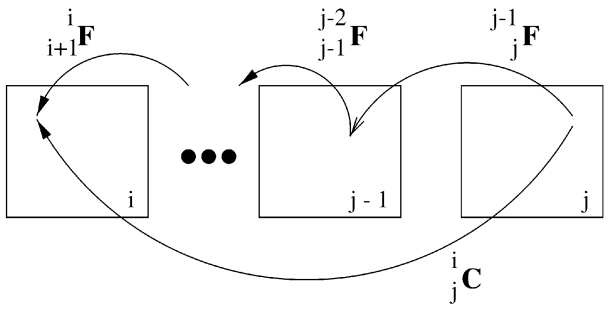

Fig. 2. Given perfect frame-to-frame flow fields $\mathbf{F}$, the theorectical cumulative flow field $\mathbf{C}$ would be identical to that obtained by composing the individual frame-to-frame flow vectors.

$${}^i_j\mathbf{C} = \begin{cases} {}^i_j\mathbf{F} & \text{if } j = i+1 \\ \Delta_j + warp({}^i_{j-1}\mathbf{C}, {}^{j-1}_j\mathbf{F}) & \text{if } j > i+1 \end{cases}, \quad (2)$$

where Δ_j is the contribution to the cumulative flow of the frame-to-frame flow from frame $j-1$ to frame j. Theoretically, Δ_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. Given perfect flow fields and perfect image warping, the cumulative flow field could be used to determine the distance each pixel has traveled from its location in a reference image. In practice, however, flow fields always contain errors. Flow fields computed on moving vegetation are particularly error-prone due to occlusions and nonrigid appearance changes. As a result, the cumulative flow vector for an image point in vegetation rarely remains correct as time goes on and more flow fields are accumulated. The cumulative flow field, therefore, does not enable scene points on vegetation to be accurately tracked, and therefore the invariant that those scene points will lie in some small bounded area over time does not yield an invariant in image space. In other words, one cannot expect that the cumulative flow field will take on only bounded values at image points that lie on oscillating vegetation. We have verified this empirically.

Although the idea of accurately measuring cumulative flow is overly optimistic, the idea behind it can be used to define a related but more appropriate measure. This new measure attempts to estimate the distance that each point travels in a consistent direction, resetting this estimate to 0 when direction changes occur. For points on objects that are easily tracked, this measure will be directly related to our definition of salient motion, presented in Section 1. For points on vegetation, if a tracking error occurs, it is highly likely that another, inconsistent, motion will pass through the image point soon, and hence the distance measure will be reset to zero. As a result, points where tracking breaks down will not take on large distance estimates.

We will now develop this measure in more detail and relate it to salience. Our desired cumulative measure will be a vector field over the image, and will be denoted $\mathbf{S}_j$. It 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, 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 (2), using the extended flow field:

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

The second and third steps detect locations that have reversed direction and reset their salience to zero. Detecting direction reversals is nontrivial, 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).[2]

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

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

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:

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

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

The y-component of the salience measure, Sj, 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 0.1, indicating that if the cumulative flow drops to 90 percent 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.

3.2 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:

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

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.

2. 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.

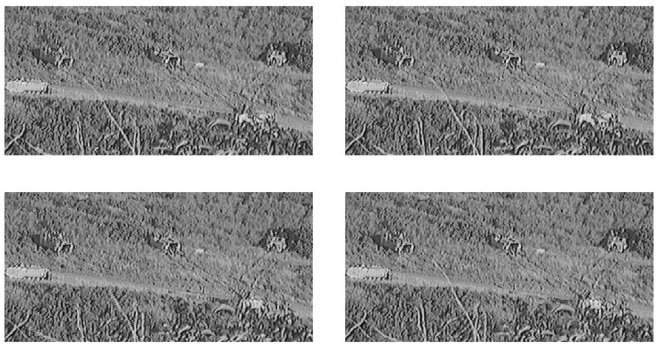

Fig. 3. Four frames I_j from a challenging video sequence, temporally ordered left-to-right, top-to-bottom.

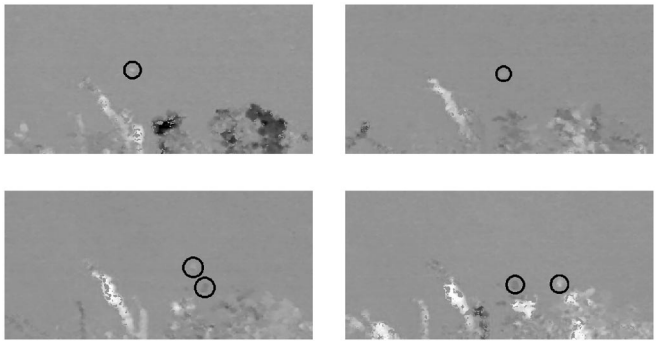

Fig. 4. X-component of the frame-to-frame flow for the frames in Fig. 3, after zeroing of those flow vectors that fail the forwards-backwards check. This is denoted ${}_j^{j-1}F_x$. This is signed imagery, so medium gray represents 0. The flow regions corresponding to the salient objects (people at a distance) have 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.

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:

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

A possible alternative to extending flow vectors, proposed by a reviewer, is to search directly for correspondences in previous frames, i.e., to compute optical flow at more than one temporal scale. This approach is not unreasonable and should be investigated further. The biggest challenge in developing such an approach is that as the frame separation becomes longer, the difficulty and computational expense of computing accurate flow grows substantially.

4 Empirical Studies

Figs. 3, 4, 5, and 6 illustrate 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. Four selected frames from the sequence, separated temporally by about 40 frames, are shown in Fig. 3. Fig. 4 shows the x-component of the frame-to-frame flow for these frames. The regions corresponding to the salient objects (people at a distance) have been circled. The frame-to-frame velocity of the people varies from 0.6 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. Fig. 5 shows the

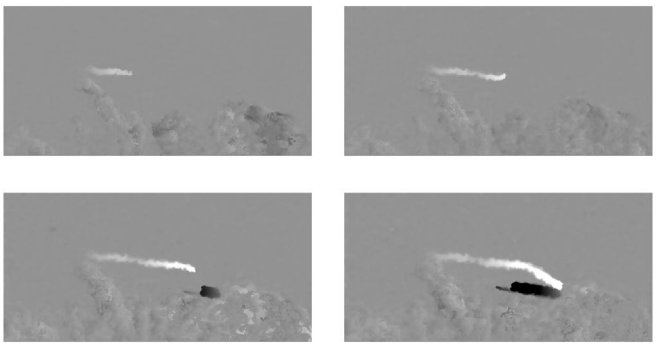

Fig. 5. X-component of cumulative consistent flow, $S_{j,x}$ for the four frames in Fig. 3. Medium gray represents 0.

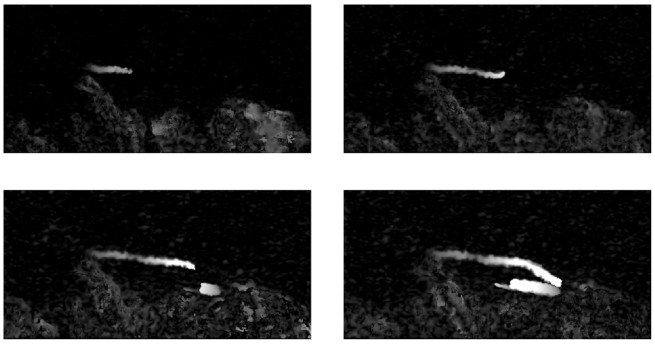

Fig. 6. Magnitude of salience, $\| \mathbf{S}_j \|$, for the four frames in Fig. 3.

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 Fig. 6. A graph showing the relative values of salience magnitude on the topmost moving object and the background points is shown in Fig. 7. In that graph, the solid line shows the average salience of points on the object. The dashed line shows, for each frame, the salience of the image point in the frame that has the largest salience but is not on the object. This can be thought of as the maximal background salience. (For comparison to the graph's temporal axis, the four frames shown in Figs. 3, 4, 5, and 6 are frames 37, 70, 113, and 150.)

As can be seen from the graph, the maximal background salience does not grow over time, but the salience of the moving object increases steadily. In the leading frames (i.e., prior to frame 50), the salience increases slowly compared to the actual distance traveled by the object (115 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 50, however, the object increases slightly in size and flow can be more reliably computed, so salience increases directly in proportion to the distance traveled.

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

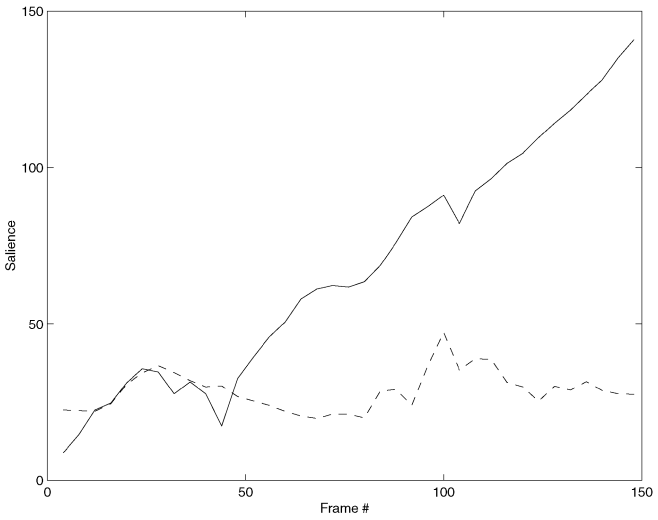

Fig. 7. Salience magnitudes for sequence shown in Fig. 6, as a function of time (frame number). Solid line shows average salience of image points on object. Dashed line shows salience of the image point with maximum salience, over all points lying outside the object.

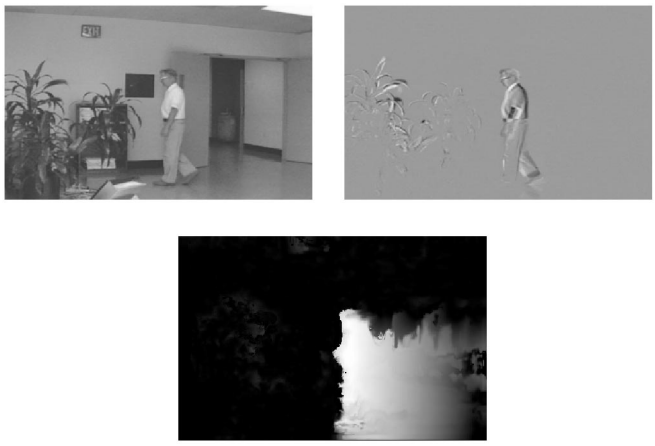

Fig. 8. Top left: Sample image from video sequence. Top right: Frame-to-frame difference. Bottom: Salience magnitude, $\| \mathbf{S} \|$.

object is the person's shadow on the ground. (The salience of this object is not plotted in Fig. 7.)

Figs. 8, 9, and 10 provide more examples of the algorithm on three other sequences. *Identical algorithm parameters were used for all four sequences.* The graphs in Fig. 11 plot the salience on the moving object and on the background in the same manner as described previously, for the sequences in Figs. 9 and 10. In all cases, the salience of the moving object rises steadly while the maximum background salience remains bounded. In Fig. 8, a person walks right to left while a fan blows the leaves of a potted plant at the left side of the image. In the frame shown, the person has traveled approximately 150 pixels and the typical salience of a pixel on the person is 140.

In Fig. 9, a person walks upper-left to lower-right against a background of gently-swaying tree branches. Because the branches are swaying only gently, the maximum background salience is relatively flat over time.

In Fig. 10, 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. In the graph for this example in Fig. 11, the salience of the person increases slowly

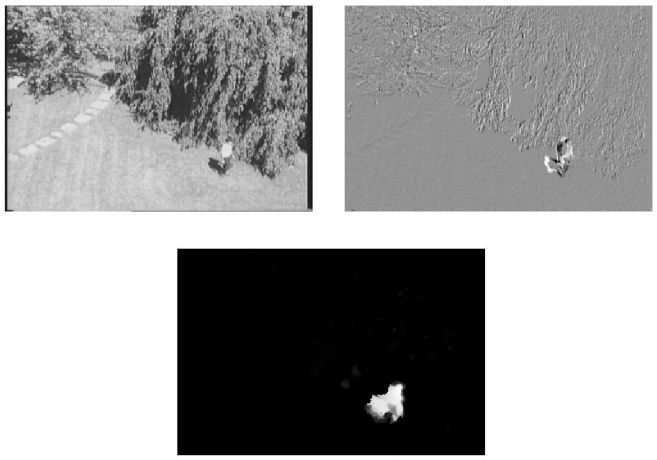

Fig. 9. Top left: Sample image from video sequence. Top right: Frame-to-frame difference. Bottom: Salience magnitude, $\| \mathbf{S}_j \|$, after suppressing the salience trails, as described in Section 5.

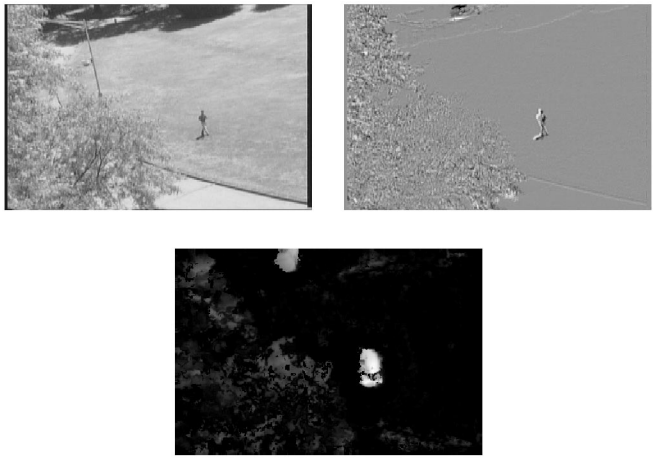

Fig. 10. Top left: Sample image from video sequence. Top right: Frame-to-frame difference. Bottom: Salience magnitude, $\| \mathbf{S}_j \|$, after suppressing the salience trails, as described in Section 5.

at first because the the person is walking towards the camera and hence the image distance traveled grows more slowly.

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 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 Figs. 9 and 10). However, there are some scenarios where it does not suffice. Consider, for example, an intruder crawling slowly beneath

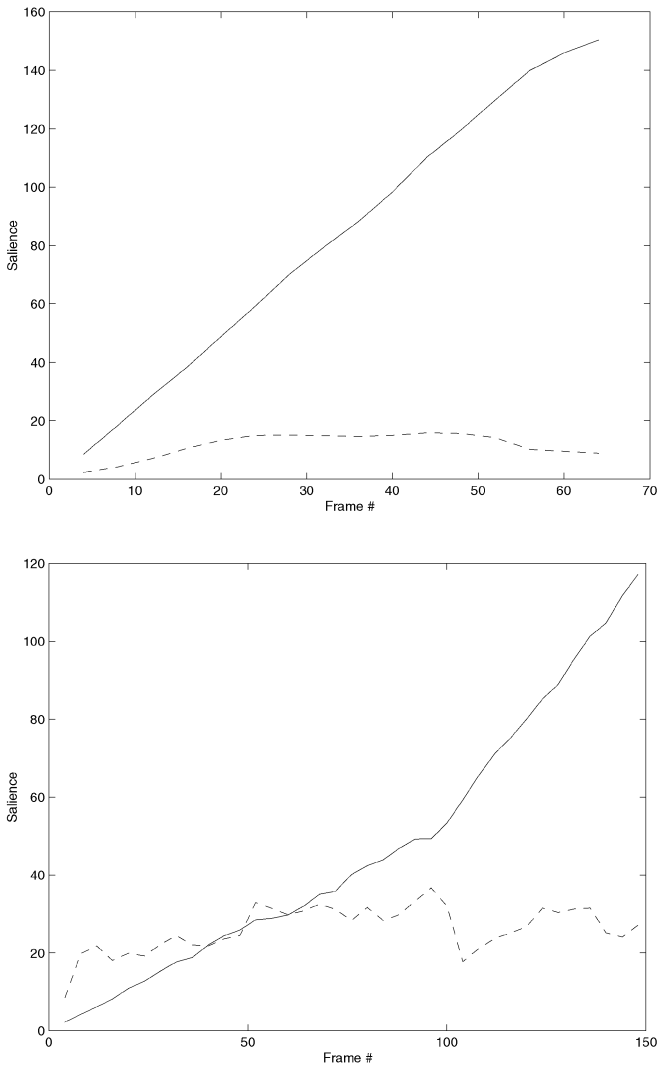

Fig. 11. Salience magnitudes for the sequence shown in Figs. 9 (top) and 10 (bottom). Solid line shows average salience of image points on object. Dashed line shows salience of the image point with maximum salience, over all points lying outside the object.

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 [9] 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. The basic algorithm relies heavily on calculating optical flow and warping images using flow vectors. However, aside from these components, the basic algorithm can be defined using point-wise pixel operations. The extended flow algorithm requires some more neighborhood search, but is not strictly required for image sequences where the targets are larger than 15-20 pixels per side. This suggests that the basic algorithm can be performed essentially as fast as one can compute optical flow. Multi-resolution gradient-based optical flow can be decomposed into spatial filtering and vector warping operations. Hardware acceleration for both of these operations is becoming widely available. Therefore, a real-time implementation of salience is quite feasible in the near future.

A key question is whether spurious flow values due to image noise can accumulate to form spurious points at which the salience is large. It is reasonable to assume that, on each frame, the x- and y-components of flow due to image noise can be modeled as Gaussians, and hence that a process that sums these Gaussians is a standard Brownian motion process. For such a process, the probability distribution function of the distance that a particle will travel in a certain time t has an expected value of 0 but a variance that is proportional to t [27]. This suggests that such accumulations would theoretically be possible were it not for resets based on direction reversal. In practice, we have not observed this accumulation. This may be due in part to resetting based on direction reversal, but the much more significant cause, we believe, is that the cumulative measures are attenuated due to the bilinear warp that is being used to warp the cumulative flow for each new frame. Such warps perform inexact interpolation that is well-known to diminish the signal being warped. Alternative approaches to preventing noise from accumulating would to combine or select from flows computed over multiple temporal scales, as mentioned above. Also, it might be possible to alter the salience measure to use unit vectors, which would have a fixed variance when summed [5].

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 [31].

ACKNOWLEDGMENTS

This work was performed while the author was with Sarnoff Corporation, Princeton, New Jersey. Thanks to Mike Hansen, Jayan Eledath, Bob Collins, Rick Wildes, and Peter Burt for providing valuable feedback.

REFERENCES

[1] C.H. Anderson, P.J. Burt, and G.S. van der Wal, "Change Detection and Tracking Using Pyramid Transform Techniques," *SPIE, Intelligent Robots and Computer Vision,* vol. 579, pp. 72–78, 1985.
[2] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. European Conf. 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," *Proc. 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," *Proc. IEEE Workshop on Motion,* Mar. 1989.
[5] R. Collins, "Multi-Image Focus of Attention for Rapid Site Modeling," *IEEE Conf. Computer Vision and Pattern Recognition,* 1997.
[6] J. Craig, *Introduction to Robotics: Mechanics and Control.* Addison-Wesley, 1989.
[7] R. Cutler and L. Davis, "View-Based Detection and Analysis of Periodic Motion," *Proc. Int'l Conf. Pattern Recognition,* 1998.
[8] G.W. Donohoe, D.R. Hush, and N. Ahmed, "Change Detection for Target Detection and Classification in Video Sequences," *Proc. ICASSP,* pp. 1,084–1,087, 1988.
[9] C. Eveland and K. Konolige, "Background Modeling for Segmentation of Video-Rate Stereo Sequences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1998.
[10] P. Fua, "A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features," *Machine Vision and Applications,* vol. 6, no. 1, 1993.
[11] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using Adaptive Tracking to Classify and Monitor Activities in a Site," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1998.
[12] G. Halevi and D. Weinshall, "Motion of Disturbances: Detection and Tracking of Multi-Body Non-Rigid Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1997.
[13] Y.Z. Hsu, H.H. Nagel, and G. Rekers, "New Likelihood Test Methods for Change Detection in Image Sequences," *Computer Vision, Graphics, and Image Processing,* vol. 26, pp. 73–106, 1984.
[14] P. Kahn, "Local Determination of a Moving Contrast Edge," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 7, pp. 402–409, July 1985.
[15] P. Kahn, "Integrating Moving Edge Information Along a 2D Trajectory in Densely Sampled Imagery," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1988.
[16] K.P. Karmann, A.V. Brandt, and R. Gerl, "Moving Object Segmentation Based on Adaptive Reference Images," *Proc. Signal Processing,* pp. 951–954, 1990.
[17] J.M. Letang, V. Rebuffel, and P. Bouthemy, "Motion Detection Robust to Perturbations: A Statistical Regularization and Temporal Integration Framework," *Proc. Int'l Conf. Computer Vision,* 1993.
[18] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving Target Classification and Tracking from Real-Time Video," *Proc. Workshop on Applications of Computer Vision,* 1998.
[19] J.J. Little and W.E. Gillett, "Direct Evidence for Occlusion in Stereo and Motion," *Proc. European Conf. Computer Vision,* 1990.
[20] F. Liu and R. Picard, "Finding Periodicity in Space and Time," *Proc. Int'l Conf. Computer Vision,* 1998.
[21] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo," *Proc. DARPA Image Understanding Workshop,* 1981.
[22] F. Meyer and P. Bouthemy, "Region-Based Tracking Using Affine Motion Models in Long Image Sequences," *CVGIP: Image Understanding,* vol. 60, no. 2, Sept. 1994.
[23] S. Niyogi and E. Adelson, "Analyzing and Recognizing Walking Figures in xyt," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1994.
[24] R Pless, T. Brodsky, and Y. Aloimonos, "Independent Motion: The Importance of History," *IEEE Conf. Computer Vision and Pattern Recognition,* 1999.
[25] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive Background Estimation and Foreground Detection Using Kalman Filtering," *Proc. ICAM,* pp. 193–199, 1995.
[26] K. Skifstad and R.C. Jain, "Illumination Independent Change Detection for Real World Image Sequences," *Computer Vision, Graphics, and Image Processing,* vol. 46, pp. 387–399, 1989.
[27] P. Todorovic, *Probability and Its Applications.* Springer-Verlag, 1992.
[28] J. Weber and J. Malik, "Robust Computation of Optical Flow in a Multi-Scale Differential Framework," *Proc. Int'l Conf. Computer Vision,* 1993.
[29] R. Wildes, "A Measure of Motion Salience for Surveillance Applications," *Proc. IEEE Int'l Conf. Image Processing,* 1998.
[30] R. Wildes and L. Wixson, "Detecting Salient Motion Using Spatiotemporal Filters and Optical Flow," *Proc. DARPA Image Understanding Workshop,* 1998.
[31] L. Williams and K. Thornber, "A Comparison of Measures for Detecting Natural Shapes in Cluttered Backgrounds," *Proc. European Conf. Computer Vision,* 1998.
[32] G. Wolberg, *Digital Image Warping.* IEEE CS Press, 1992.
[33] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, July 1997.
[34] S. Yalamanchili, W.N. Martin, and J.K. Aggarwal, "Extraction of Moving Object Descriptions Via Differencing," *Computer Graphics and Image Processing,* vol. 18, pp. 188–201, Feb. 1982.