Determining Temporal Persistence and
Consistent Edge Motion From Natural Images

Philip Kahn

COINS Technical Report 87-56

June 1987

ABSTRACT

Objects do not randomly appear and disappear from images. Rather, they appear, move, and then disappear from view. This paper introduces two related measures of temporal continuity in images: temporal persistence and consistent edge motion. *Temporal persistence* is the period of time over which a spatial feature has stably persisted in the image plane. *Consistent edge motion* occurs when an edge moves in a manner which is consistent with its previous trajectory. These new temporal features are important to object recognition because spatial structure which persist in a stable manner over time are more likely to be structurally related to objects. This paper describes how these new features can be computed from natural imagery. An event-driven method for determining edges and their normal motion extends previously reported work. As in all low-level vision models, the computation places constraints on the variation of the intensity surface. Because natural images conform to the less restrictive laws of physics, they inevitably contain intensity variations which violate model assumptions. Fortunately, previous image motion provides low-level context that can determine whether current motion measurements are the result of modelled intensity variations. This observation is used to develop a technique for determining temporal persistence and consistent edge motion using past motion and local neighbor communication. The approach is demonstrated on natural imagery.

This research was supported in part by the Defense Advanced Research Projects Agency under contract number N00014-82-K-0464 and monitored by the Office of Naval Research.

Determining Temporal Persistence and Consistent Edge Motion from Natural Images[1]

Philip Kahn

Computer Vision Research Laboratory
Department of Computer and Information Science
University of Massachusetts, Amherst

Abstract:

Objects do not randomly appear and disappear from images. Rather, they appear, move, and then disappear from view. This paper introduces two related measures of temporal continuity in images: temporal persistence and consistent edge motion. *Temporal persistence* is the period of time over which a spatial feature has stably persisted in the image plane. *Consistent edge motion* occurs when an edge moves in a manner which is consistent with its previous trajectory. These new temporal features are important to object recognition because spatial structures which persist in a stable manner over time are more likely to be structurally related to objects. This paper describes how these new features can be computed from natural imagery. An event-driven method for determining edges and their normal motion extends previously reported work. As in all low-level vision models, the computation places constraints on the variation of the intensity surface. Because natural images conform to the less restrictive laws of physics, they inevitably contain intensity variations which violate model assumptions. Fortunately, previous image motion provides low-level context that can determine whether current motion measurements are the result of modelled intensity variations. This observation is used to develop a technique for determining temporal persistence and consistent edge motion using past motion and local neighbor communication. The approach is demonstrated on natural imagery.

[1]This research was supported in part by the Defense Advanced Research Projects Agency under contract number N00014-82-K-0464 and monitored by the Office of Naval Research.

I. Introduction

The world around us has remarkable continuity. A car moving down a street cannot instantly reverse direction. A ball dropped from a building can be expected to fall to the ground. A moving object occupies a space and moves into an adjacent space. Without this inherent continuity, our world would be a jumbled collage of images.

The inherent continuity of environmental motion provides strong expectations about the structure of time-varying imagery. Objects do not randomly appear and disappear from images. Rather, they appear, move, and then disappear from view. This paper introduces two related measures of temporal continuity in images: temporal persistence and consistent edge motion.

Temporal persistence is the period of time over which a spatial feature has stably persisted in the image plane. For example, consider an edge which appears, moves and then disappears from the image plane. The temporal persistence of the edge at any moment is the amount of elapsed time since the edge first appeared in the image.

Temporal (dis)occlusion is related to temporal persistence. Temporal *disocclusion* occurs when a spatial feature newly appears in the image (i.e., the feature has no previous temporal persistence). Conversely, temporal *occlusion* occurs when a spatial feature disappears from the image (i.e., the feature ceases to temporally persist). Thus, the detection of temporal (dis)occlusion is implicitly required to determine temporal persistence.

The way spatial features change over time suggests their importance to the object recognition task. *Spatial structures which persist in a stable manner over time are more likely to be structurally related to objects.* Conversely, temporally transient spatial features are more subject to noise and other factors which are not related to object structure.

Consistent edge motion occurs when an edge moves in a manner which is consistent with its previous trajectory. For example, a newly disoccluded edge does not have a previous trajectory, and thus it cannot be consistent with previous image motion. Conversely, an edge constantly translating across the image is always consistent with its previous trajectory.

Because edges describe a small local portion of a spatial image, edge detectors are somewhat noise sensitive. Put another way, edge detectors extract edges which may or may not have a structural relationship to physical objects in the environment. Though this point is somewhat apparent, it deserves discussion. Metaphorically, intensity surfaces are much like the surface of the ocean. Some changes in the surface are due to waves. Other changes may be due to wind, chop, etc. If we are measuring waves (which are much like lines in images), then other factors such as

wind and chop constitute noise for the "wave extraction process." Statistically, the noise factors can spatially and locally appear "wavelike," though these factors lack the coherence in time of the wave. It is the coherent and consistent movement of a wave in time which clearly distinguishes it from other unrelated factors. Similarly, it is the coherent and smooth movement of lines (and constituent edges) in time-varying imagery which distinguishes meaningful edges from noise; this coherent and smooth motion of edges is called consistent edge motion.

The next section overviews the event-driven computation of temporal persistence and consistent edge motion. Event-driven edge detection in natural images is then discussed. The subsequent section describes the low level tracking of edges used to ascertain temporal persistence and consistent edge motion. Results on a natural image sequence and a general discussion follows.

II. Computational Overview

A. Computational models of time-varying imagery

Time-varying imagery (TVI) may be described by a three dimensional *spatiotemporal cube* [4] in which all points are defined by (x, y, t) where (x, y) defines a point in the bounded image plane and t is a point in time. The time domain is defined by the variations along the "column" of this spatiotemporal cube (i.e., variations at a pixel over time). Similarly, the space domain is defined by the variations of intensity on the image plane at an instant of time.

Computational models for time-varying imagery (TVI) may be categorized by noting whether they discretize the space and time domains[2]. A domain which has a discrete formulation should be considered continuous if either formulation is plausible. Since each domain can be either discrete or continuous, there are four possible ways to distinguish among TVI models: implementational models (discrete space and time), physics models (continuous space and time), frame-based models (continuous space and discrete time), and time-based models (discrete space and continuous time).

Implementational models assume in their formulation that both space and time are discrete. Since physical devices limit changes in space and time, actual algorithm implementation occurs at this level. Although many engineering models fall into this category, we know of no computational models which *require* both discrete space and discrete time.

Physics models describe physical phenomena independent of the sensor by relying upon principles from physics (e.g., fluid mechanics [40], optics [17], and signal theory [12]). These models

[2]P. Anandan of Yale University contributed to the development of this taxonomy.

examine the continuous formulas governing variations of the image in both space and time. Since they tend to be analytically rigorous, these models often allow definite statements to be made about the inherent nature of the visual task. Additionally, the physics analogy allows many established tools to be used in model formulation (e.g., Fourier analysis, ray-tracing, and regularization). Though a powerful analytic technique, mapping a physics model into a computation can be difficult, and the resulting computation is often combinatoric and/or noise sensitive.

Frame-based models consider time-varying imagery as a sequence of frames in time; the frames can be viewed as "slices" in the spatiotemporal cube. These models are by far the most prevalent in the literature (e.g., [3,6,23,27,28]) since current video and film technology captures images as a sequence of frames. It is thus the best understood of the computational frameworks.

Frame-based techniques sample the spatiotemporal cube at discrete points in time. Since objects move continuously in space and the velocity at which they move can assume a wide range of values, a feature in one frame can in principle move anywhere in the continuous space of the next frame (including off the bounded space of the image frame). Determining where a feature has moved between two frames has been termed the *correspondence problem* [37]. Determining the correspondence of features over the "gap" of time between frames is made difficult because no inherent constraint on the spatial displacement exists, though weak heuristics usually provide unique and efficient solutions [27]. A more general extension of the correspondence problem is the *multiple frame integration problem* which seeks to track image features over multiple frames in order to attain greater accuracy [6,28].

Time-based models use the fixed spatial relationships among image positions to interpret the order of temporal image events. The earliest versions were presented as models of biological retinas (e.g., [29]), but recently there has been a renewed interest with a focus towards computer vision (e.g., [1,19,32,33,38,39]).

Time-based models have asynchronous flow of information and they naturally allow parallel computation. These models are generally defined by local automata which are data driven and respond asynchronously to image events. This approach to computation is analogous to biological information processing, which accounts for its early use as a model of retinal motion detection.

B. Review of time-based models

When discussing image velocities, it is important to explicitly specify the type of velocity. Figure 1 illustrates a moving edge. The *real* motion of the edge is represented by V_R. Due to the *aperture problem,* a straight edge viewed in a small local region will always appear to move in a direction

perpendicular to itself [37]. V_N in figure 1 represents the *observed* motion which is always normal to the brightness gradient (also called *normal* motion). The aggregate of real velocities within an image is often called the *optical flow* [15,37], and its determination requires the integration of locally observed velocities [3,17].

Time-based models of imagery were first advanced as models of biological motion perception. Reichardt first proposed a class of motion detection models which this paper refers to as *opponent correlation* models [29]. As shown in figure 2a, Reichardt's motion detector contains two subunits that are attuned to opposite directions. Each subunit multiplies the input from two point receptors (one of which is passed through a linear temporal filter (TF) to approximate a delay) which is then temporally averaged. Leftward motion is indicated when the output of the right subunit exceeds that of the left subunit; similarly, rightward motion is indicated when left subunit output exceeds right subunit output. As shown by van Santen and Sperling [33], the original Reichardt model can suffer from a form of aliasing that causes incorrect output. Their elaborated Reichardt detector (ERD) avoids this potential problem by adding additional spatial filters (SF in figure 2b) and incorporating a few simple assumptions; [32] notes that the ERD is fully equivalent to the detector proposed by Adelson and Bergen [1], and for suitably chosen filters it is equivalent to the detector proposed by Watson and Ahumada [39].

A somewhat related approach to motion perception has defined spatiotemporal filters which respond optimally to a range of velocities [12]. These approaches are inspired by the *velocity-tuning* found in biological vision pathways [13] and this class of motion detectors can be referred to as *velocity-tuned filter* models. These models observe that the temporal frequency $\mathcal{F}_T$, the spatial frequency $\mathcal{F}_S$, and real image velocity V_R are related by $\mathcal{F}_T = \mathcal{F}_S V_R$. The space and time frequency domains are then bandpass filtered in order to obtain a velocity-tuned filter. This can be demonstrated using figure 3, which shows a contour plot of temporal frequency expressed as a function of spatial frequency and real velocity (the isoclines indicate temporal frequency). The spatial bandpass is indicated with a heavy dashed line, and the temporal bandpass is indicated with a heavy solid line; the resulting spatiotemporal filter only passes input which is within the hatched area shown in figure 3. This spatiotemporal filter is velocity-tuned in that it does not pass input which has real velocity outside the range indicated with a dotted line in figure 3; though the shaded area in figure 3 is within this velocity range, it is not passed by the filter (hence, velocity-*tuning* vs. bandpass).

The approach proposed by the author in [19] explicitly detects and tracks continuously moving

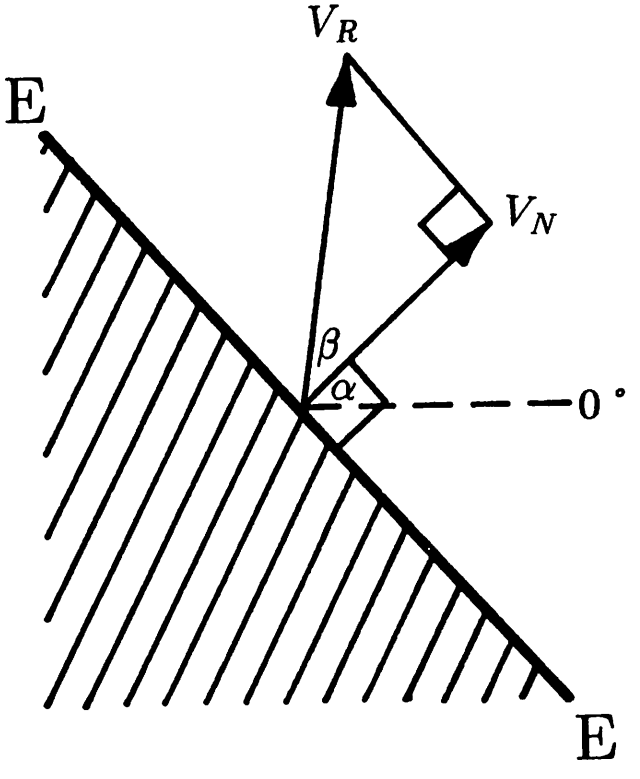


Figure 1: Geometry of a moving edge and the *aperture problem*.

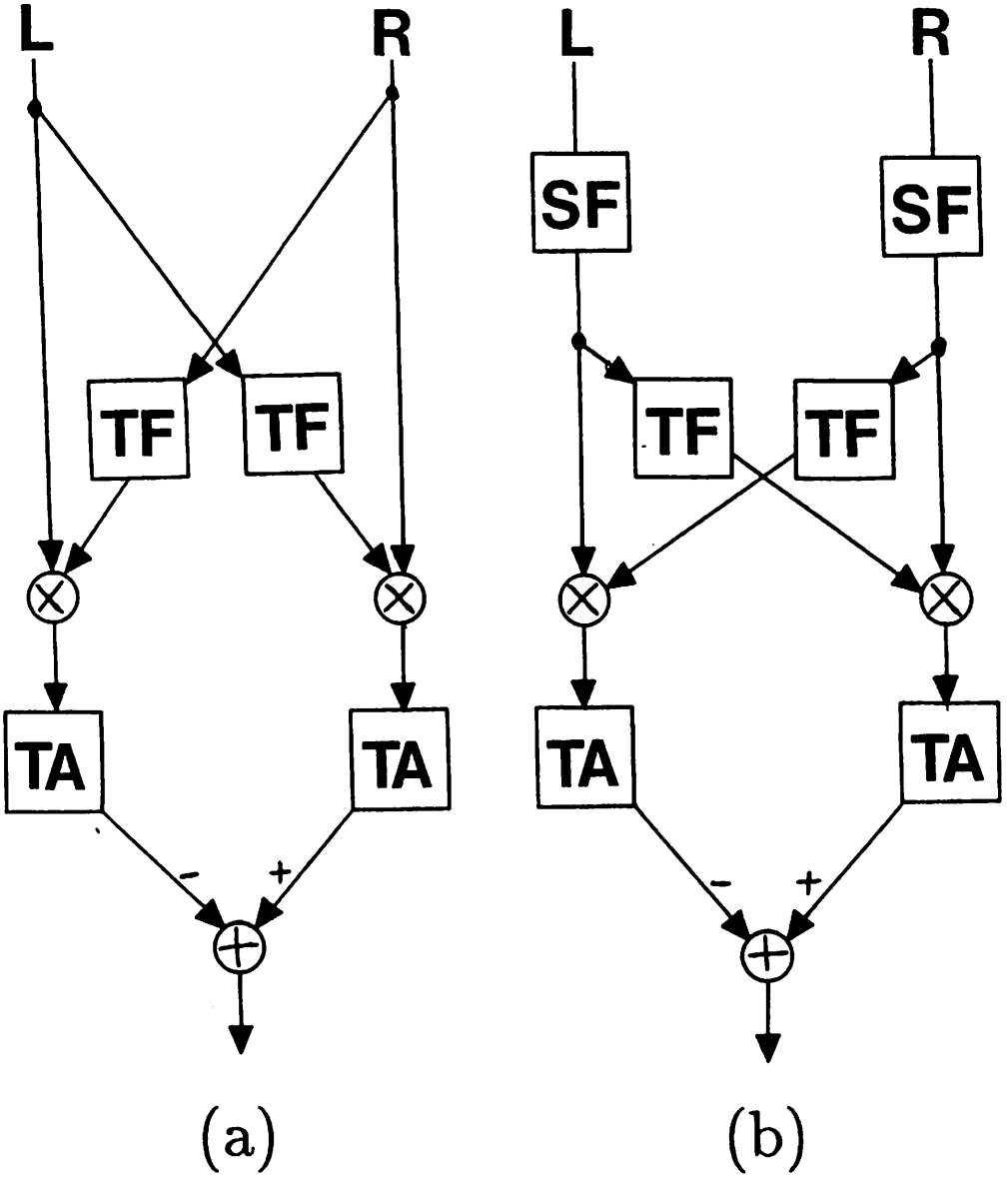


Figure 2: (a) Reichardt's original motion detector [29]; (b) elaborated Reichardt detector [32]. *TF* denotes a linear temporal filter, × indicates a multiplication, *TA* denotes time averaging, and + indicates an algebraic addition.

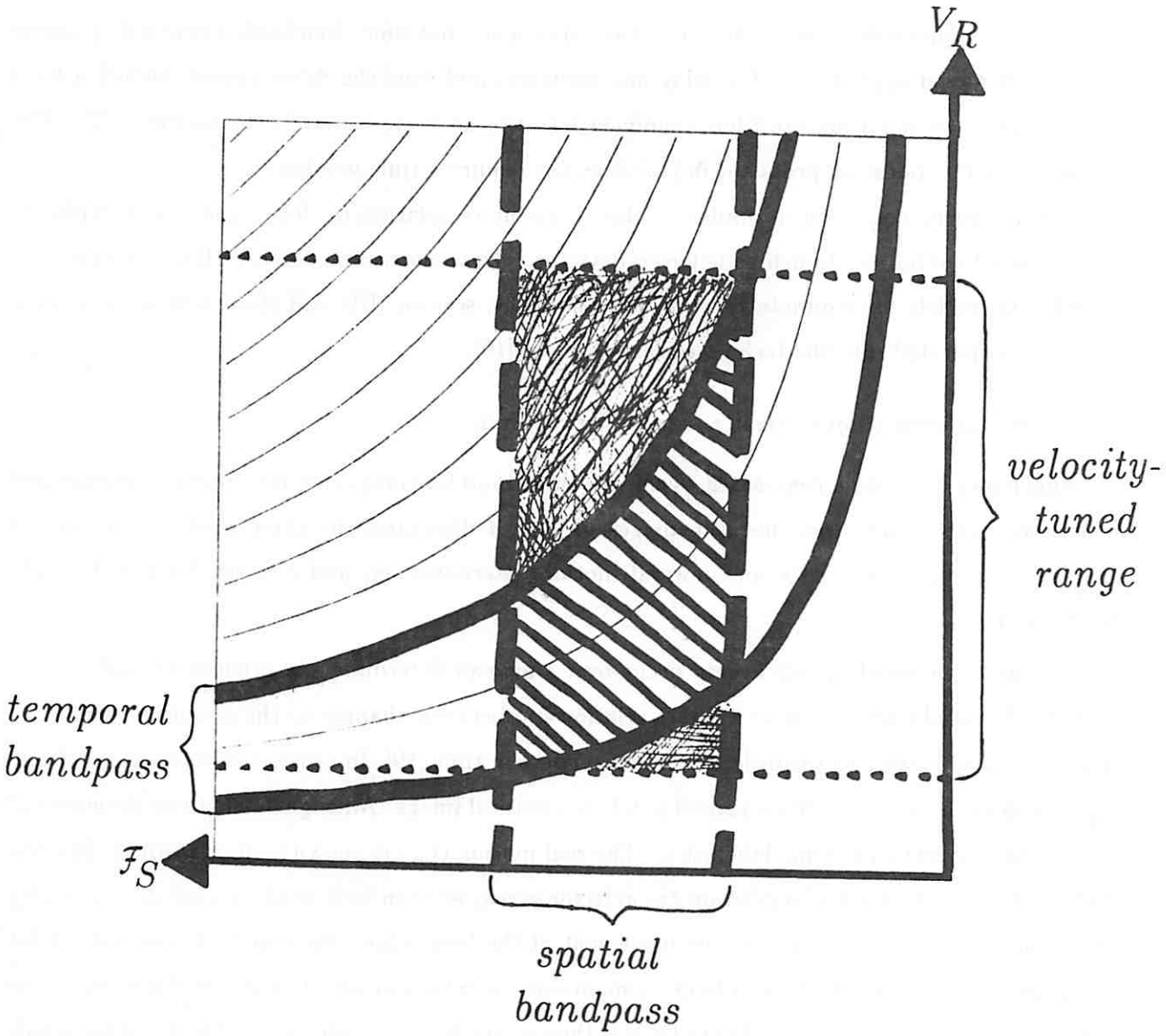


Figure 3: A contour plot of temporal frequency $\mathcal{F}_T$ expressed as a function of spatial frequency $\mathcal{F}_S$ and real velocity V_R: $\mathcal{F}_T = \mathcal{F}_S V_R$. The isoclines indicate temporal frequency.

edges. An edge was defined as a locally straight contrast boundary. As an edge moves relative to the detector array, portions of the edge traverse pixels; a pixel "event" occurs the moment a portion of the edge traverses the pixel. A locally straight edge causes these events to occur in a fixed and precise order in time which allows edge orientation and normal motion to be determined.

The velocity-tuned filter models constrain the time domain by operating within a localized time window. Though significant events may occur at many temporal scales, spatiotemporal filters are velocity-tuned precisely *because* they are localized in space and time. Reichardt's original opponent correlation model approximated a delay and compare, and thus the delay period defined a fixed time window; more recent work has examined how this restriction may be overcome [1,32]. The event-driven computation proposed in [19] does not require a time window.

The model in [19] is most similar to the opponent correlation models. They both explicitly use the fixed spatial relationships between detector cells to determine motion. Because opponent correlation models discriminate between left and right, section IIID will show how these models can be incorporated into the technique developed by [19].

C. Computational stages: events, edges, consistency

This paper develops a *time-based,* event-driven method for computing temporal persistence and consistent edge motion from natural images. Figure 4 illustrates the three main computational stages: event detection, edge and normal motion determination, and context-dependent model enforcement.

The order in which image events occur over time can determine the orientation and normal motion of spatial edges; there is a direct relationship between changes in the spatial location of a moving straight edge and the order of temporal edges in time [19]. In order to develop an intuition, figure 5 shows a local edge from a small patch in a natural image. An edge in the three dimensional projection is marked by a line labelled E. The real motion (V_R) is known from subsequent images. *Taking the vantage point of a pixel on the detector array,* we then look at the variation of intensity at this pixel over time caused by the movement of the local edge. As a first approximation, let us assume the intensity surface in figure 5 maintains its form and simply translates with constant velocity. The intensity at a pixel over time is thus a "track" in the direction of real motion which has the width of a single pixel. Three such tracks are shown in figure 5. In this restricted case, it is apparent that the moving spatial edge E results in a one-dimensional edge measured in the time domain which occurs when the spatial edge is centered over the target pixel (i.e., where the edge E intersects a pixel "track"); this can be clearly seen in figures 5 and 8. Note that temporal

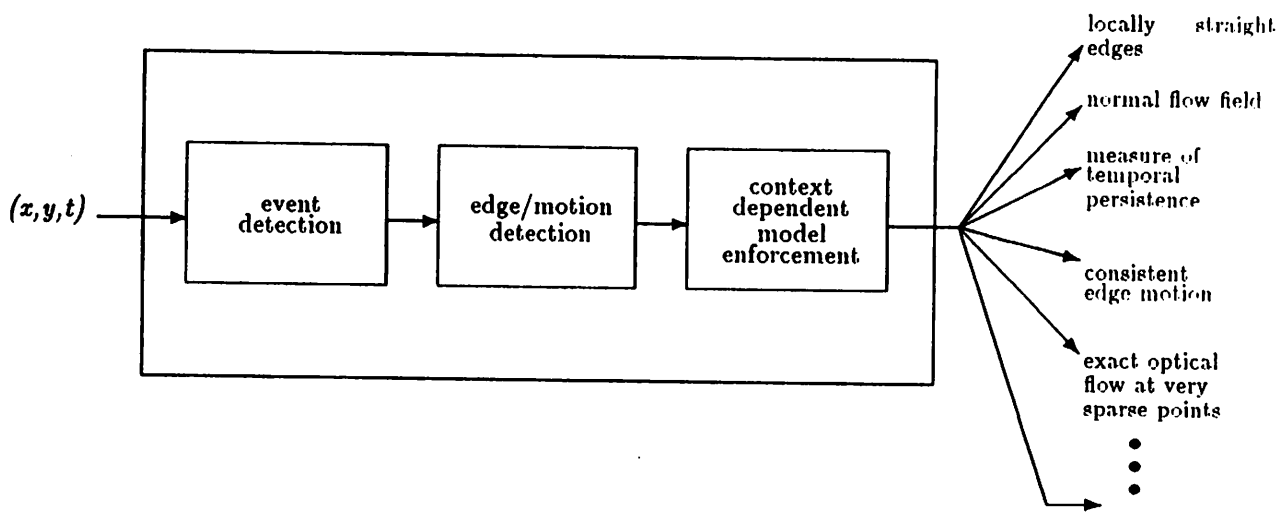


Figure 4: Input/process/output view of event-driven computational model.

edges detected from different pixels retain the spatial structure of the two-dimensional edge E. Temporal edges are detected in time, they are thus extracted using an *event detector* (the first stage of processing in figure 4). Determination of edge direction and motion is the second stage in figure 4.

To simplify the discussion, it was initially assumed that the intensity surface in figure 5 rigidly translated with constant velocity. This allowed the spatial edge E and the pixel intensity values over time (i.e., pixel "tracks") to be intermingled in figure 5. This assumption need not be retained. Nonconstant velocity compresses and expands the temporal signal relative to the spatial signal (i.e., real velocity and temporal frequency are directly related), so temporal edges can retain their structural relationship to spatial edges. Intensity variations due to sensor bias and fluctuation, aliasing, and other effects can undermine the structural relationship between temporal and spatial edges; section IV discusses how these effects can be overcome.

This section has provided a very general overview. The next section describes how moving edges can be extracted from natural, time-varying imagery. Section IV shows how consistency among adjacent edge/motion detectors can determine temporal persistence and consistent edge motion. This is followed by a presentation of results and a discussion.

III. Event-Driven Edge Detection

Figure 6 shows the steps that can be used to determine the orientation and normal motion of edges in natural, time-varying imagery as proposed in [19]. Acquisition, initial filtering, and detection of temporal edges (i.e., *event* detection) are first discussed. The detector geometry described in [19] is then extended to arbitrary tessellations.

A. Low-pass spatial filtering

Camera lenses generally pass spatial frequencies significantly higher than the maximum sampling rate which is justified by the resolution of the detector array [42]. To avoid sensor aliasing, a spatial low-pass filter (LPF) can be used to ensure that no frequencies are passed which are higher than that justified by the size of the pixel [38]. To be most effective, sensor sampling should occur *after* the spatial LPF (as shown in figure 6).

The edge detection model proposed in [19] which is discussed in this paper assumes that no more than one locally straight edge may simultaneously traverse an edge motion unit; this may be called the ***singularity assumption***. Thus, if w is the maximum distance between any two pixels contained

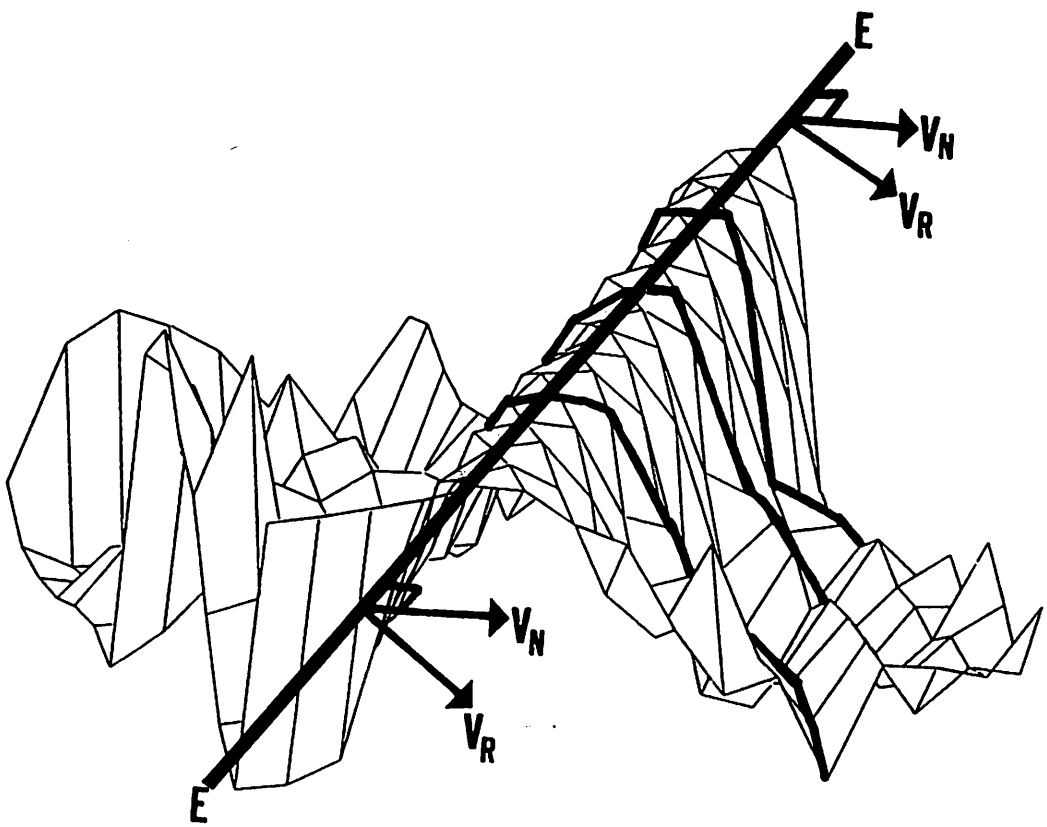


Figure 5: Translating straight edge from a small patch of a natural image.

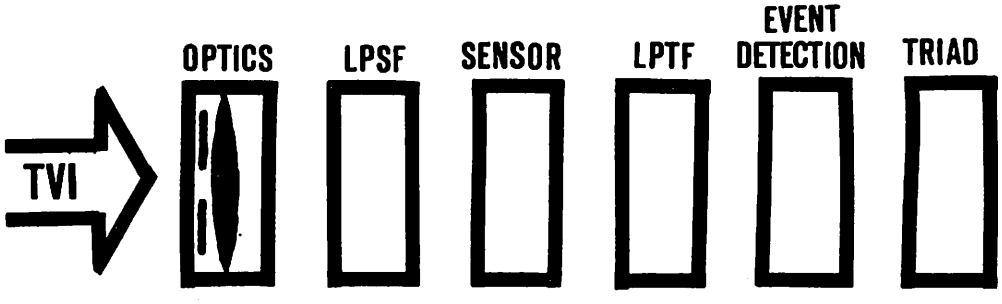


Figure 6: Steps that can be used to determine orientation and normal motion of edges in natural imagery (per [19]).

within a single motion unit, the singularity assumption requires that the minimum distance between any two edges be no less than w. This may be accomplished by low-pass spatial filtering, though the required cutoff frequency depends upon the adopted definition of a locally straight edge. For example, if edges are defined by local maxima, then $w = 1/B$ where B is the spatial cutoff frequency of the low-pass filter [22].

Due to the central limit tendency of natural phenomena [21,31], defocusing, Lambertian reflectors, objects out of the depth of field, and a great many other optical effects result in gaussian smoothing [42]. Because gaussian smoothing attenuates higher frequencies, it serves as a reasonable low-pass filter [8]; static optical elements can thus provide inexpensive, reasonable quality low-pass spatial filters that can operate before sensor sampling. Alternative spatial low-pass filters which better approximate the ideal are discussed in [22,26,31,42].

B. Low-pass temporal filtering

Image sensors are an aggregate of discrete spatial detectors which individually sample intensity over time. The pixels in a spatial image must be sampled over the same time interval in order to preserve *temporal coherence*. Not all image sensors preserve this property. For example, raster-scanned video tube cameras sample each pixel at a different time; because the pixels imaged from a moving object will not all be measured at the same time, the sensor does not preserve temporal coherence. Conversely, motion pictures and many CCD cameras preserve temporal coherence because the elements in each frame are exposed over the same time interval.

An imaging device only approximates the optical time-varying image because the response of phototransducers is affected by cell composition, load and pre-historic conditioning, light level, etc. [2]; these effects acts as a low-pass filter in the time domain [2,35]. Sampling the spatiotemporal cube in discrete slices of time (i.e., frames) usually results in temporal aliasing because the temporal LPF cutoff determined by phototransducer response characteristics is generally significantly higher than the temporal frequency cutoff dictated by the frame sampling rate (e.g., 30 frames per second). As a result, additional low-pass temporal filtering to match frame rate is usually needed to reduce the effect of temporal aliasing.

The implementation of a low-pass temporal filter differs somewhat from that of a spatial filter. Specifically, low-pass spatial filters are generally implemented by some symmetric smoothing function (e.g., a gaussian). A symmetric smoothing function in the time domain seems to present a problem, since one cannot smooth into the future (as implied by a symmetric gaussian with mean at the current point in time). This paradox can be resolved by setting limits on the smoothing

function and moving the origin to the upper limit (which delays the output by half the range of the bounded function). Because a temporal smoothing function can only be applied to signals measured in the past, asymmetric functions seem more appropriate than symmetric functions; they also require less delay than their symmetric counterparts. Figure 7 shows some asymmetric functions which were considered; the experiments presented in this paper used a normalized half gaussian as a temporal low-pass filter.

As noted before, $\mathcal{F}_T = \mathcal{F}_S V_R$. One interesting consequence of this equation is that a family of spatial frequencies and real velocities can result in the same temporal frequency (as shown by the isocontours in figure 3). This *temporal equivalency* effect is discussed by Watson and Ahumada in their derivation of the *critical sampling frequency* [39].

Closer examination of figure 3 shows that filtering in one dimension (i.e., temporal frequency, spatial frequency, or real velocity) may be indirectly achieved by filtering another dimension (e.g., as done in the velocity-tuned filter models). For example, increasing real velocity for a fixed temporal LPF is equivalent to decreasing the cutoff of a spatial LPF by a constant factor (i.e., as V_R increases for fixed $\mathcal{F}_T$, $\mathcal{F}_S$ decreases); this explains motion blurring caused by excessive movement of the camera relative to the environment [5,23]. Conversely, increasing the cutoff of the temporal LPF for a fixed real velocity is equivalent to increasing the maximum spatial frequency by a constant factor (i.e., as $\mathcal{F}_T$ increases for fixed V_R, so does $\mathcal{F}_S$). In this sense, low-pass spatial filtering can be equivalently accomplished by appropriate temporal low-pass filtering. When real velocity V_R and the temporal LPF cutoff are known, the highest passed spatial frequency is known. In general, only the temporal LPF cutoff is known when filtering occurs; real velocity is computed by later stages. In this case, increasing the temporal LPF cutoff is equivalent to increasing the highest passed spatial frequency, though the precise relationship is not known. This can be useful when a shift in spatial frequency is desired, since there are practical limits to the dynamic modifiability of the low-pass spatial filter and sampling rate.

C. Detecting events from natural time-varying imagery

As shown in figure 5, there is a relationship between spatial edges and temporal edges (i.e., *events*). This subsection examines how these events can be detected in a one-dimensional temporal signal (i.e., pixel intensity over time).

Any technique for detecting an "event" over the temporal signal is valid as long as it tends to detect events that are part of a larger, locally straight contrast edge. There are many signal events that can be defined by simple variations in the low-order derivatives of a signal; this section will

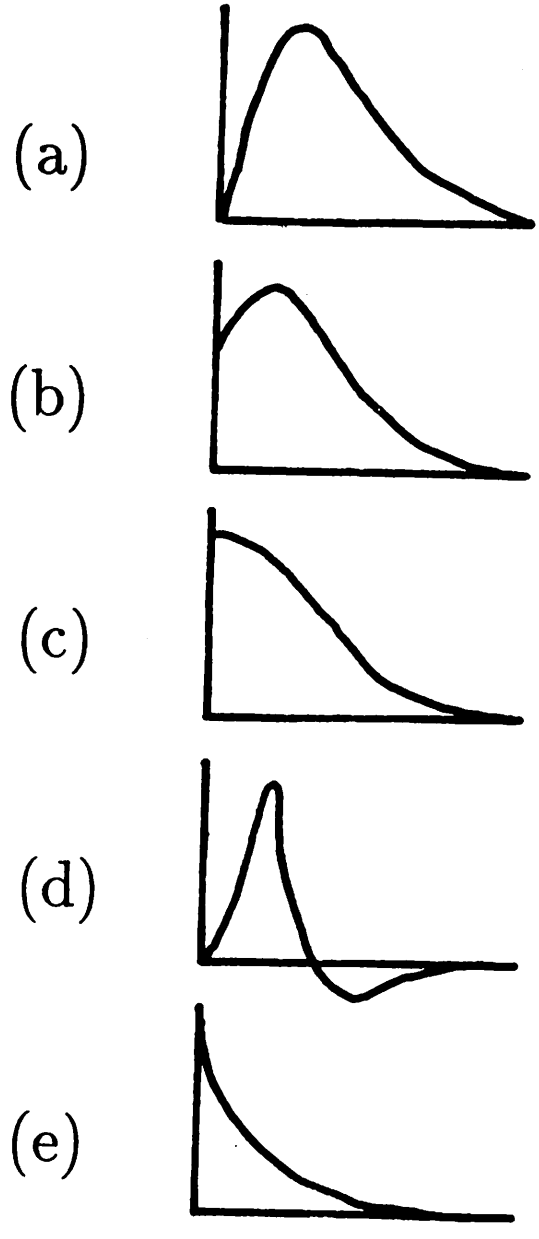


Figure 7: Some temporal low-pass filter functions: a) Poisson function, b) neural model (Tyler in [36]), c) half gaussian, d) neural physiology (Adelson and Bergen in [1], e) exponential decay.

discuss the use of minima, maxima, and inflection points for temporal edge extraction.

The extrema of a one-dimensional signal can be simply detected by noting when a change in sign occurs in the first derivative of the signal. When the sign of the first derivative goes from positive to negative, a maximum in the signal has occured. Conversely, when the sign of the first derivative changes from negative to positive, a minimum in the signal has occured. Inflection points are simply detected by noting when the second derivative changes sign.

Figure 8 shows the intensity at a pixel over time. This is analogous to the pixel "track" shown in figure 5. Temporal filtering has been done to avoid aliasing due to frame-based image acquisition. Minima (N), maxima (M), and inflection points (I) are indicated.

When an image moves with nonconstant velocity, there is no longer a direct correspondence between units of time and space (i.e., the temporal signal is not a scaled version of the spatial signal in the direction of real motion). Rather, portions of the spatial signal are relatively expanded and compressed in the temporal signal by changes in velocity. This compression/expansion does not affect the number of detected minima/maxima. Because extrema which form a contour generally have similar velocity, nonconstant velocity tends to uniformly displace them so that the local two-dimensional structure of the edge is preserved.

Though extrema provide a useful definition of a temporal edge, inflection points are generally considered to be the most common definition of an edge; edges are usually thought to separate regions of relative lightness and darkness. This more accepted definition may be incorporated by defining temporal edges as inflection points in the temporal signal. With constant velocity, temporal inflection points structurally relate to spatial edges much the same as temporal extrema. Nonconstant velocity causes the temporal signal to expand or compress relative to the spatial signal; this can introduce or eliminate inflection points in the temporal signal. That is, nonconstant velocity undermines the structural relationship between inflection points in the temporal and spatial signals. When the assumption of locally constant velocity is incorporated, the correspondence of temporal inflection points to spatial inflection points is ensured.

The model of a translating locally straight edge does not depend upon a particular event definition. Multiple event definitions (e.g., minima *and* maxima) can provide denser edge and normal motion fields, though only one event type may be used at any one time due to the singularity assumption.

It is important to note that event detectors are defined by whether they tend to detect portions of a translating straight edge. This section has only examined minima, maxima, and inflection points as potential events; other techniques may prove superior by demonstrating a higher correlation

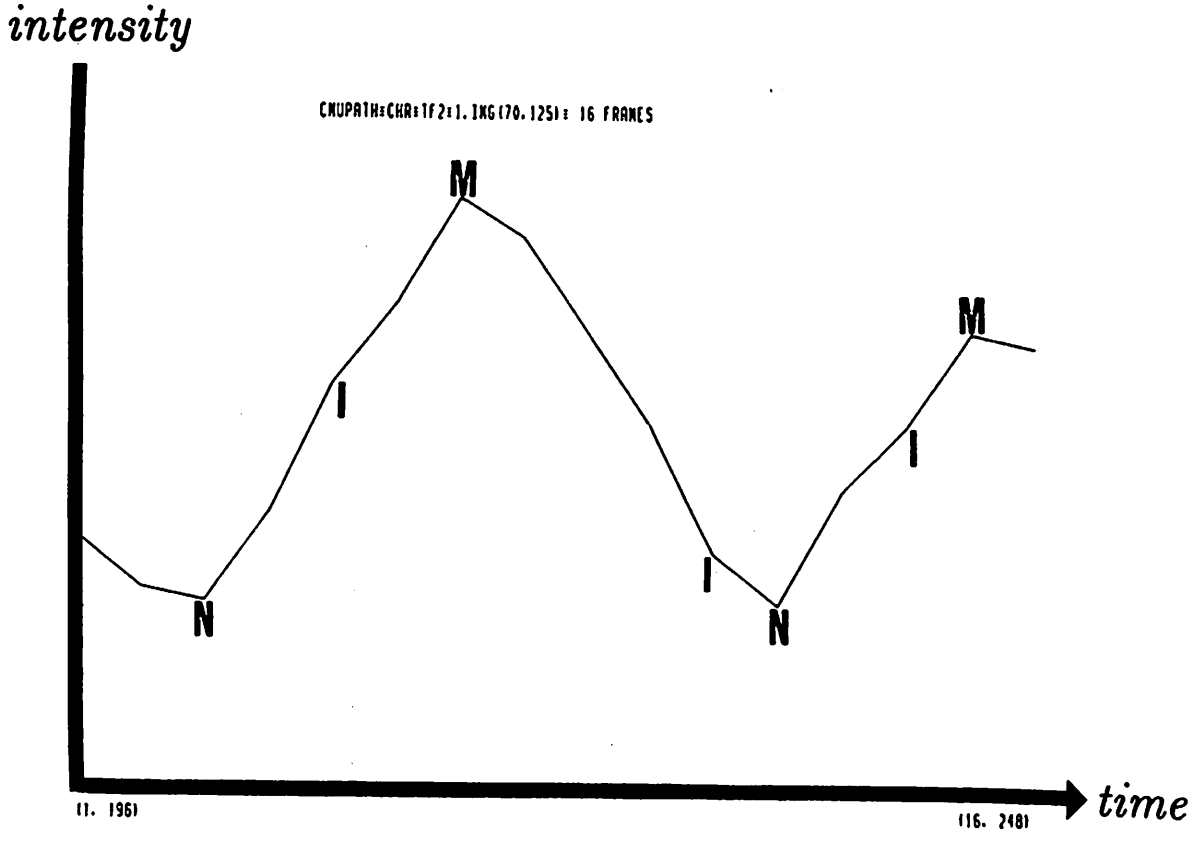


Figure 8: Intensity at a pixel over time as the image patch translates.

between detected events and their membership in a moving locally straight edge.

D. Event-driven edge detection

A time-based computational model for determining the spatial orientation and normal velocity of a moving edge was presented in [19]. The technique is based upon the model of a two-dimensional straight edge continuously translating over a fixed tesseral unit. Because a minimum of three noncollinear points are required to determine the general motion of a line on a plane, this fixed tesseral unit is called a *triad*. Each cell in the *triad* is an event detector (as described in the previous subsection) which responds when a portion of a locally straight edge traverses it. The fixed geometry of the cells in a detector unit and the well-defined model of a moving edge allows space-time relationships to be determined analytically. In short, these relationships allow the orientation and normal motion of a moving edge to be determined by the order and relative timing in which the triad cells are traversed in time. This section extends the analysis in [19] to arbitrary detector tessellations; see [19] for a more complete discussion of the technique.

Because most sensors used for computer vision use a square detector tessellation, the three-celled unit described in [19] cannot be directly used for this imagery. Instead, the technique can be generalized to arbitrary event detector tessellations by observing the order in which the detectors are traversed by a moving edge. For example, figure 9a shows nine arbitrarily placed event detectors labelled A through I. Changing the direction of a moving edge can change the order in which these event detectors are traversed in time. Thus, in figure 9b, the edge moving downward and slightly to the right traverses A then B; in figure 9c, a slight change in edge direction to downward and left reverses the traversed order of event detectors.

Suppose that each event detector (e.g., A through I in figure 9a) is a node in a fully connected graph where each arc corresponds to a possible traversal order of two event detectors in time, and every possible order of event detectors in time can be represented by a path in this graph. Only a subset of these paths can be generated by a continuously translating straight edge; it always traverses the cell(s) closest to it in the direction of real motion. This natural constraint prunes the fully connected graph into a tree, and this tree is equivalent to a finite automaton which can determine edge direction based upon the traversal order of event detectors [19].

We now derive a finite automaton for square event detector tessellations which determines edge direction from the order of detected temporal edges in time (i.e., events). Figure 10a shows four event detectors arranged in a square tesseral unit (as exists in most image sensors) and the possible directions of a moving edge (in radians). Enumerating the order of event detectors for all

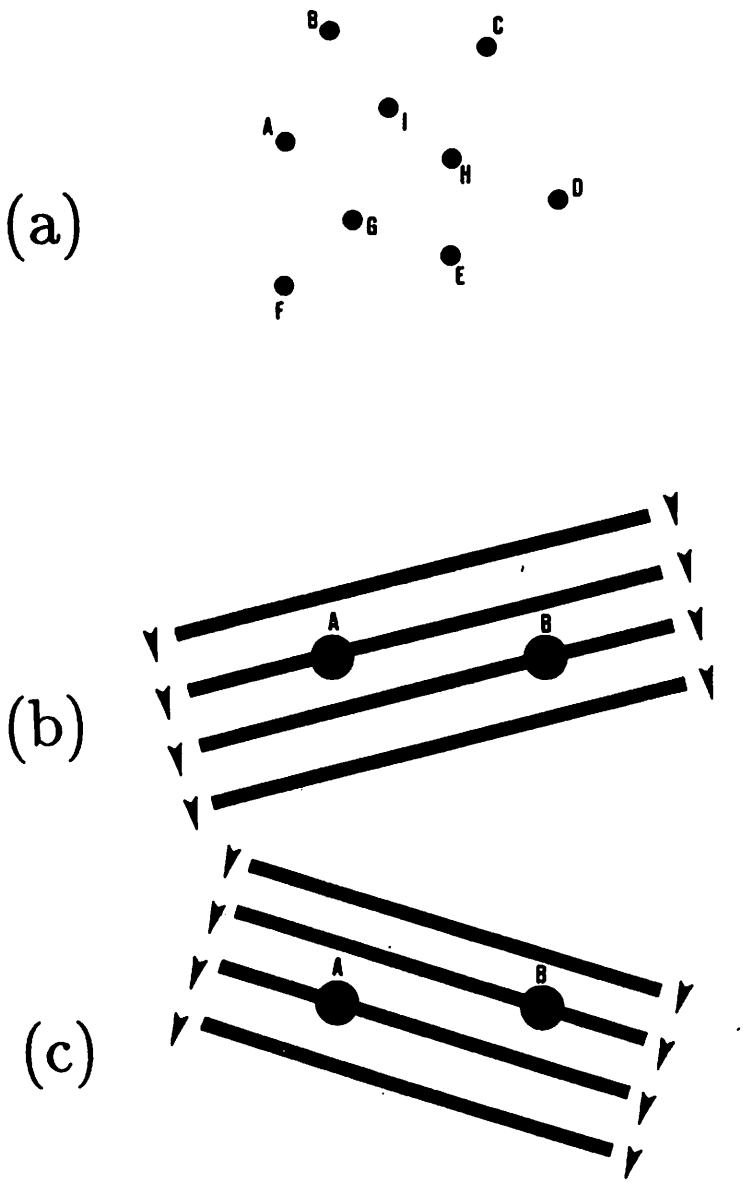


Figure 9: Extending *triads* to arbitrary event detector cell configurations.

translating edge directions yields the eight equal sectors with size $\pi/4$ radians shown in figure 10a. Each section has an *order of passage* which constrains edge direction to that sector (e.g., sector 1 has order of passage $ACBD$). When an edge traverses the first cell in a passage, six sectors are eliminated and edge direction is limited to two adjacent sectors. When the second cell is traversed by the edge, another sector is eliminated and the edge direction is limited to a single sector. Exact edge direction may be determined in a manner analogous to that in [19] when the assumption of constant velocity is incorporated.

Figure 10b shows a finite automaton representation of figure 10a which uses the traversal order of event detectors in time to determine edge direction. The initial state assumes that no event detectors have been traversed. The circles, triangles, squares, and hexagons indicate whether one, two, three, or four event detectors, respectively, have already been traversed. Arcs between nodes indicate the event detector(s) traversed at an instant in time. Figure 10b is essentially a time ordered representation of the spatial relationships shown in figure 10a.

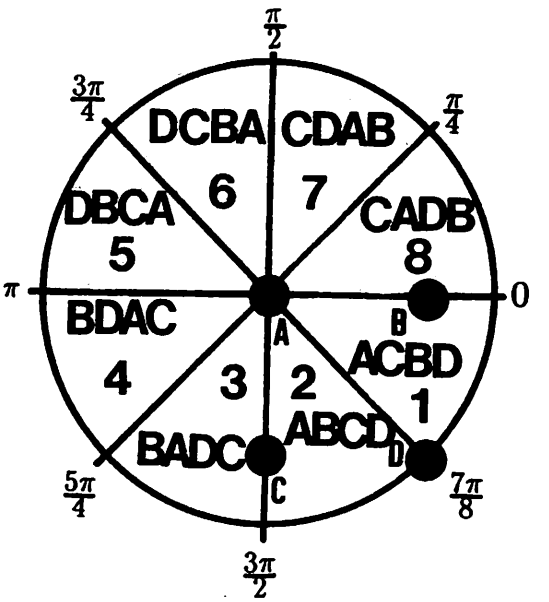
Opponent correlation models, since they determine left/right motion (analogous to the left/right motion detection shown in figures 9b and 9c), can replace the event detection technique presented in subsection C while still preserving the ability to determine the general motion of a line (as presented in this section). That is, the technique employed in this section and in [19] simply incorporates the opponent correlation models described in section IIB.

IV. Context Dependent Model Enforcement: Low-Level Tracking

The previous section presented an event-driven edge detection technique which is based upon the model of a locally translating straight edge. As in all low-level vision models, the computation places constraints on the variation of the intensity surface. When image variations occur which do not conform to a computational model, the resulting output may not have a known or correct interpretation. Because natural images conform to the less restrictive laws of physics, they inevitably contain intensity variations which violate model assumptions.

There are factors inherent to physical devices (e.g., transient noise, sensor fluctuation and response limits, optical defects, etc.) which undermine the direct correspondence between the environment and its projection onto the image plane. Figure 11 shows some other image variations which do not conform to the *triad* model. A translating edge that occludes (e.g., figure 11a) never fully completes traversing the *triad*. A disoccluding edge (e.g., figure 11b) does not fully traverse the *triad* unit, hence, the order of passage is not indicative of edge direction and motion. The

(a)



(b)

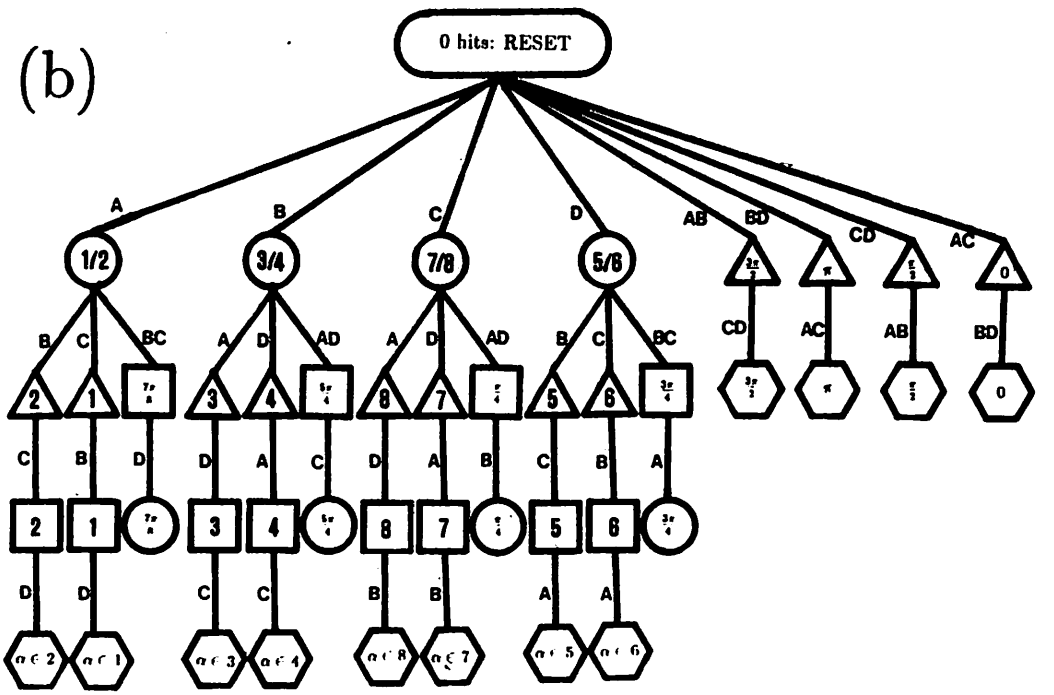


Figure 10: (a) *Order of passage* for a square *triad* unit; (b) nondeterministic finite automaton representation of a).

order of traversal created by a nonstraight local image (e.g., figure 11c) is also not indicative of edge direction/motion. In animal vision, where saccades generate motion which is a prerequisite for visual perception [20], there are often edge direction reversals whose order of passage violates the *triad* model (e.g., figure 11d). Though these model violations cannot be detected by local *triad* edge/motion detectors, they can be detected and corrected by enforcing consistency at a scale larger than single edge/motion units.

Figure 12 shows how the *triad* model may be enforced. *Event-driven edge detection,* presented in the previous section, uses detected temporal *events* (see section IIIC) and the current *triad* state and finite automaton (see section IIID) to determine the *hypothetical edge* direction and motion (as shown in figure 12). When the local intensity variation conforms to the model of a translating straight edge, the hypothetical edge correctly describes the underlying intensity variation. Conversely, when the local intensity variation does not conform to the *triad* model (e.g., see figure 11), the hypothetical edge is not meaningful. Determining the validity of the hypothetical edge is called *model enforcement.*

The validity of the edge hypothesis can be determined by examining its consistency with previous image motion. As shown in figure 12, previous image motion provides *expectations* and a current measure of *temporal persistence* which which can be used to determine whether the current edge hypothesis arose from modelled local intensity variation (i.e., a locally translating straight edge). Previous image motion provides low-level *context* that can determine the validity of the edge hypothesis; hence, the approach is called *context-dependent model enforcment (CDME).*

The next subsections describe how expectations and local neighbor communication can be used to determine the validity of the current edge hypothesis, determine *temporal persistence* and *consistent edge motion,* and provide virtually noise-free feature extraction from general imagery.

A. Propagating expectations among adjacent neighbors

Time-based models need not "lose" track of image features since there is no gap of time in which the position of the feature cannot be monitored. This introduces the inherent constraint that a moving edge which does not disappear from the image is constrained to move through adjacent edge/motion detectors.

Figure 13 shows a locally translating straight edge. The trajectory of the edge is shown. Previous positions of the edge along the trajectory are *upstream* in time; conversely, positions that the edge will occupy in the future are *downstream* in time. The boxes marked 1 – 8 in figure 13 are edge/motion detectors. Thus, the edge shown in figure 13 is currently traversing the center

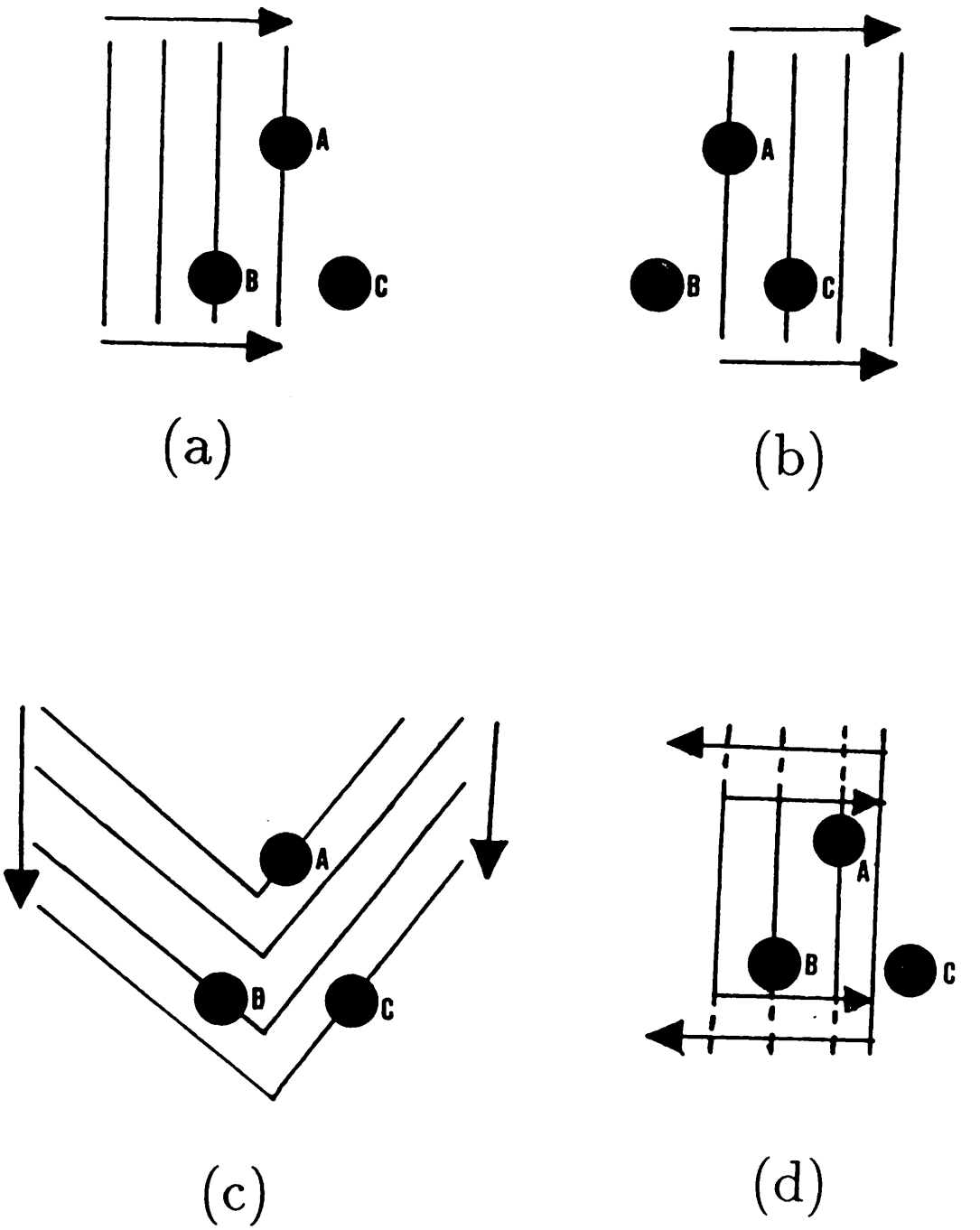


Figure 11: Image variations which do not conform to the *triad* model: a) occluding edge, b) disoccluding edge, c) nonstraight edge, d) edge direction change.

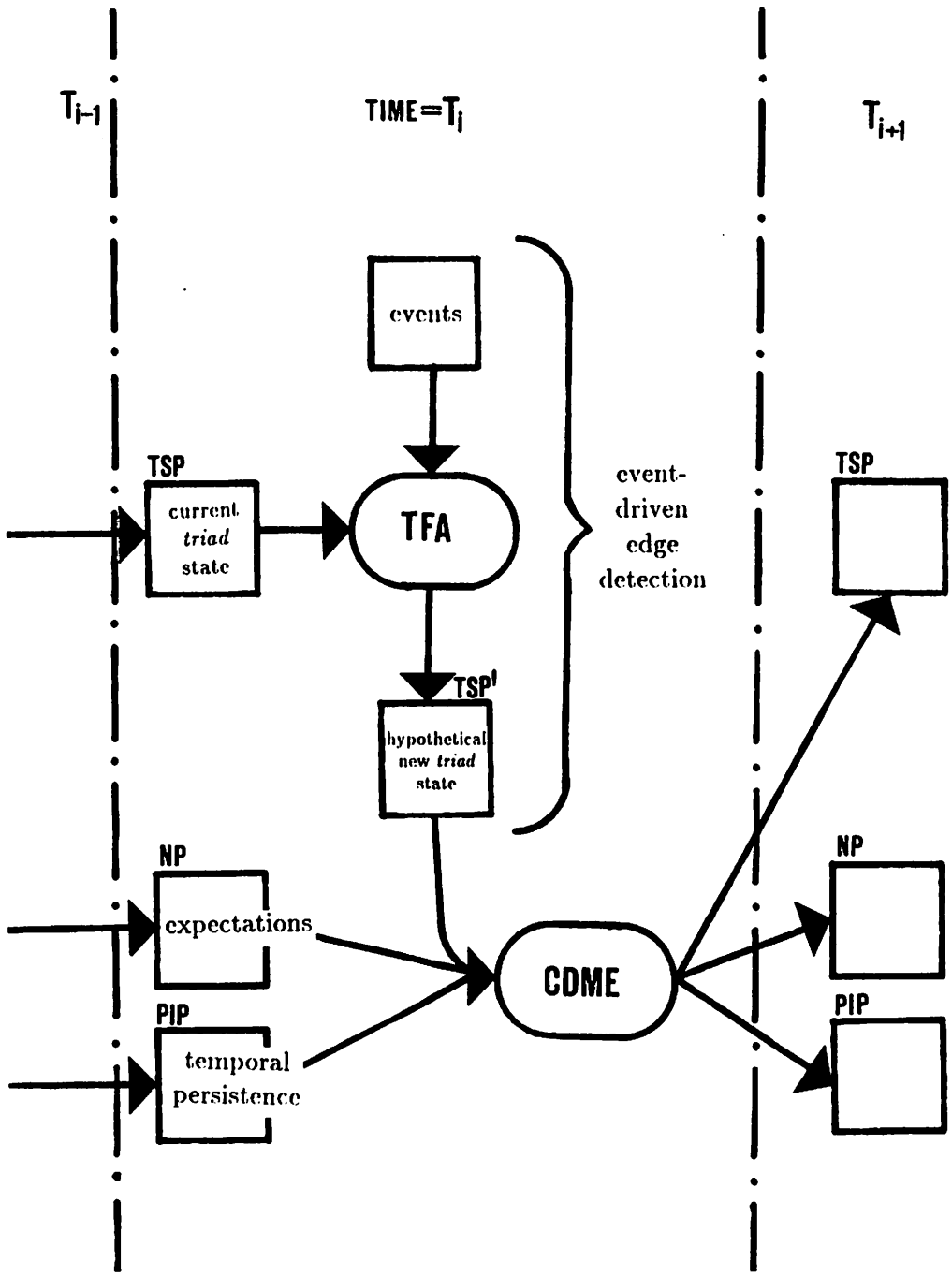


Figure 12: Block diagram of local edge/motion detection and model enforcement.

edge/motion unit, it traversed unit 7 upstream in time, and it will traverse unit 3 downstream in time.

As an edge moves, information about its motion, direction, etc., may be propagated to other edge/motion units along the trajectory. For example, in figure 13 the information available to the center edge/motion unit can be propagated downstream in time to unit 3; when the edge reaches unit 3, the expectations provided by upstream detector units can be used to determine whether the edge hypothesis is consistent with the computational model (i.e., consistent with a translating locally straight contrast edge). Similarly, the center unit may propagate its current information upstream in time to units that have already been traversed by the edge.

In general, propagating information downstream in time provides predictive expectations; these expectations can be used to determine the validity of the current edge hypothesis. Propagating information upstream in time allows more accurate feature tracking by allowing the resetting of the edge/motion state to be affected by downstream data (i.e., resetting to the zero hit state shown in figure 10b). This paper concentrates on the downstream propagation of feature information; current work is examining the use of upstream propagation for better synchronization and tracking.

If time-based image acquisition is assumed, then only communication between adjacent units is required; section IVE will show how this easily extends to frame-based image acquisition paradigms. Regardless of whether image acquisition is synchronous or fully parallel, the communication channels between edge/motion units is always fixed and localized; hence, there is no search, and communication is particularly well-suited for VLSI implementation.

Two techniques for propagating expectations along the edge trajectory were explored. In the first approach, edge information was only propagated to the single downstream unit closest to the trajectory; for example, in figure 13 the edge traversing the center unit would only propagate information to unit 3. Because inaccuracies in local edge direction often occur in natural imagery, this approach sometimes results in a failure to correctly communicate expectations along the edge trajectory. For reasons that will become apparent in subsection C, it is better to be less restrictive when communicating edge traversal expectations.

Alternatively, a *group* of adjacent units can be notified of an approaching edge. For example, in figure 13, information about the edge currently at the center unit can be propagated to units 2, 3, and 4. This additional "slop factor" allows for small local inaccuracies in edge direction that are endemic to natural imagery. It virtually ensures that consistently moving edges will correctly communicate expectations of edge characteristics to units along the trajectory. As shown in the next subsection, these expectations can be used to determine the validity of the current edge hypothesis.

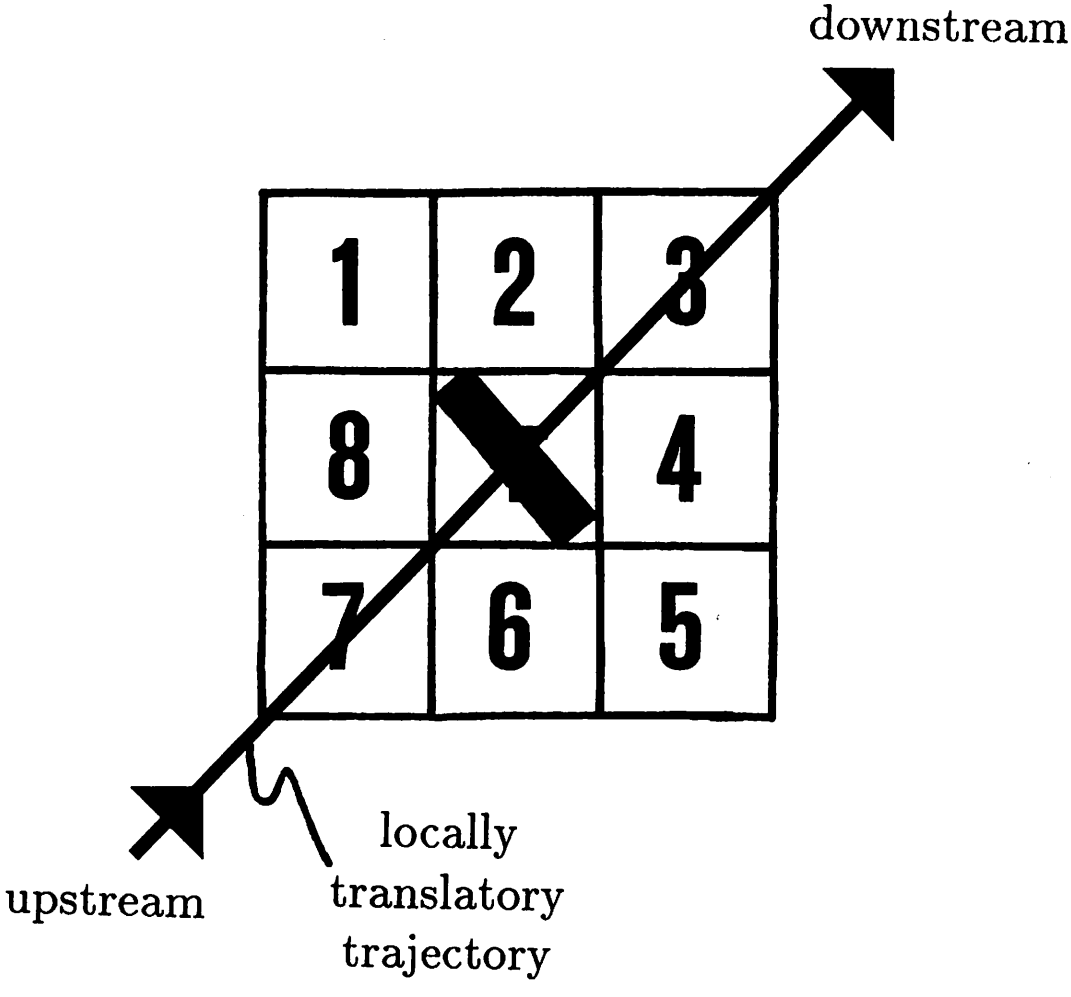


Figure 13: Propagating expectations along edge trajectory.

B. Local consistency determines edge hypothesis validity

The requirement that current image motion be consistent with previous image motion is a simple, yet powerful, way to determine the validity of the current edge hypothesis (as shown in figure 12). To provide a more intuitive sense of motion coherence, figure 14 shows the normal motion field computed from natural imagery using an event-driven edge detector as previously discussed. A six frame sequence was used[3] and the local normals for all frames have been compressed into a single image. Intuitively, consistent motion in figure 14 occurs where there seems to be common motion over time. For example, vectors which go head to tail indicate consistent motion. Conversely, there are edge vectors in figure 14 which appear almost random; these local normals do not display temporal persistence or agreement with previous image motion.

Some violations of the *triad* model (e.g., direction reversal as shown in figure 11d) are locally detectable by the edge/motion unit, but many others cannot be detected using only local information. When an edge traverses an edge/motion unit (e.g., the center unit in figure 13), the unit forms a hypothesis describing edge direction, motion, and perhaps other information. As discussed, this hypothesis is only valid when the variation in image intensity conforms to the computational model. The validity of an edge hypothesis can be largely determined by verifying its consistency with the model of a translating straight edge and the expectations provided by adjacent edge/motion units. The key idea is that when an edge hypothesis is found inconsistent, it is not incorporated as the new edge/motion detector state (in the sense of figure 10b).

The *triad* model of a translating locally straight edge implies the simultaneous existence of certain local and neighbor information. For example, to be consistent with the model, a local edge/motion unit must be notified by an upstream neighbor of an approaching edge before it is locally detected. Thus, the *disoccluding* edge shown in figure 11b is not consistent with the model because the edge newly appeared in the image, hence, there are no upstream units that could have warned of an approaching edge; the edge hypothesis for that detector is thus inconsistent with the *triad* model, therefore, the edge hypothesis should not be incorporated as the new detector state. Similarly, *nonstraight* local image patches (as in figure 11c) tend to create edge directions that vary along the trajectory of real motion; this tends to cause breaks in neighbor communication along the trajectory.

Other than notifying of approaching edges, neighbors can provide more detailed information about the moving edge to downstream neighbors (e.g., the amount of time required to completely

[3]The office pan sequence was provided courtesy of R.C. Bolles of SRI International and it is reported by Bolles and Baker in [4].

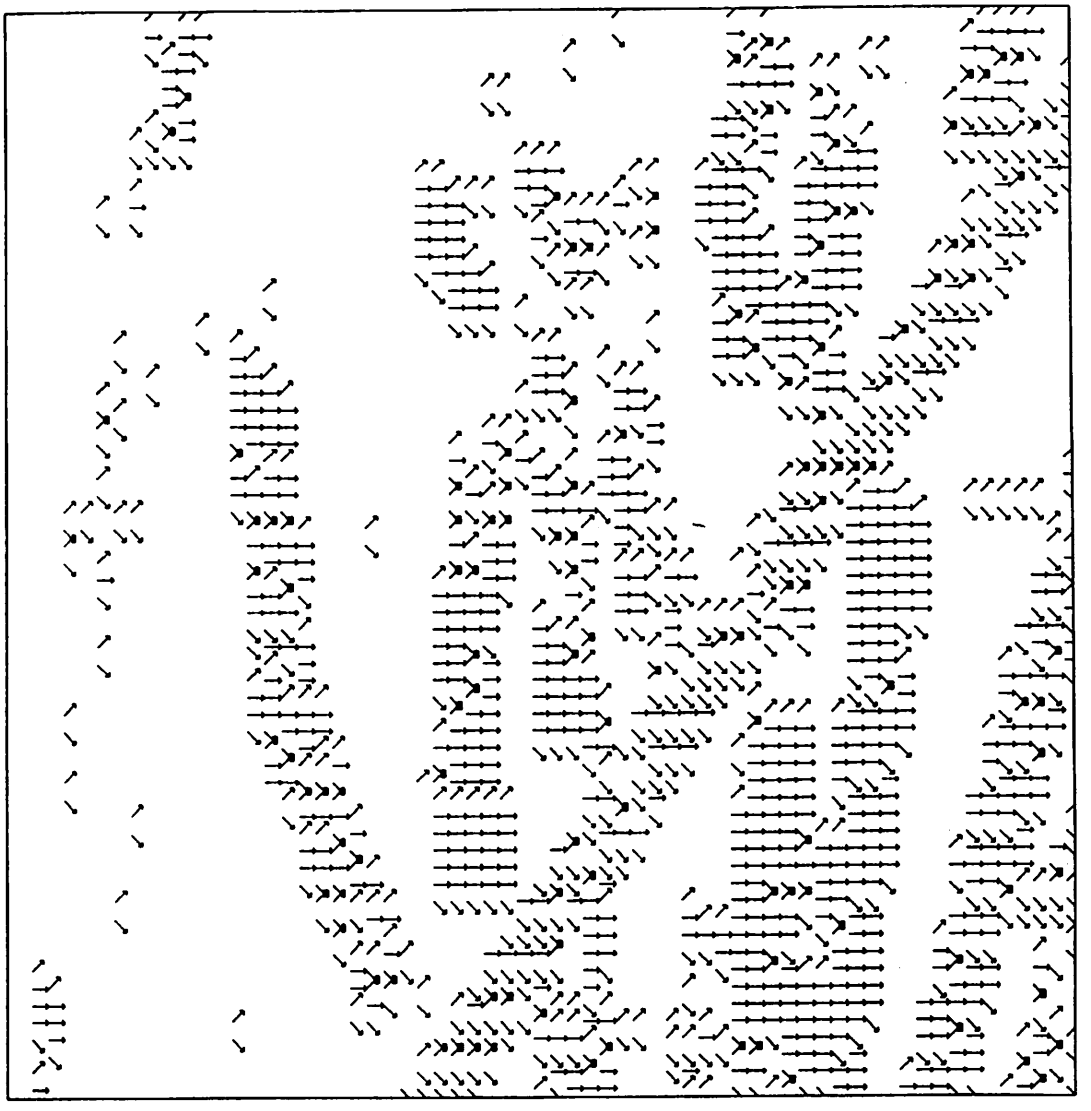


Figure 14: Local normals computed from a six-frame natural image sequence using an event-driven edge detector and compressed into a single frame.

traverse the upstream neighbor). It is important that the state of edge/motion detectors accurately reflect the structure of the underlying intensity surface. For example, the *occluding* edge shown in figure 11a leaves the edge/motion automaton in a state that indicates that two out of three detectors have been traversed. It is best to reset the state of the detector as soon as possible to reflect the disappearance of the moving edge from the image plane. If neighbor information is not employed, the only solution is to reset the detector state after a fixed time period starting from the first event detector traversal. Yet, the local edge/motion unit could take advantage of information available from upstream units to more quickly reset its state. For example, if the time required to traverse a unit were propagated downstream, that time period could be used in place of a fixed time period; this would result in a tighter relationship between the state of local detectors and the underlying intensity image.

The local consistency checking presented here was found sufficient to process natural images (as shown in section V); current work is exploring other dimensions of local and neighbor consistency checking. The next section ties the ideas of *neighbor communication* and *local consistency checking* together in order to describe how they can combine to yield virtually noise-free feature extraction.

C. Virtually noise-free feature extraction

The previous subsections have described how neighbor communication and local consistency checking can be used to enforce the *triad* model. Though these two alone are not sufficient to *always* detect and correct errors caused by unmodelled intensity variations, they have a simple extension which can virtually guarantee that extracted features are error-free.

Figure 15 defines some useful nomenclature. The communication between two edge/motion units is a *link*. As an edge moves along a trajectory, a *chain* of links is formed; for example, a chain of length three is shown in figure 15.

Assume that p is the probability that a *link* has no structural relationship to the environment; the exact value of p can be empirically estimated, though a particular value is not needed for this discussion. The key idea is that *the cumulative probability that a chain is not structurally significant is p^n where n is the chain length.* Because this cumulative probability is an exponentially decreasing function, even a relatively small chain length can virtually assure that the chain is due to modelled image variations. Thus, virtually noise-free feature extraction can be obtained by requiring that "visible" features have a minimum associated chain length.

Subsection A discussed how expectations may be propagated among adjacent neighbors. In particular, it was stated that it is better to be less restrictive when communicating edge traversal

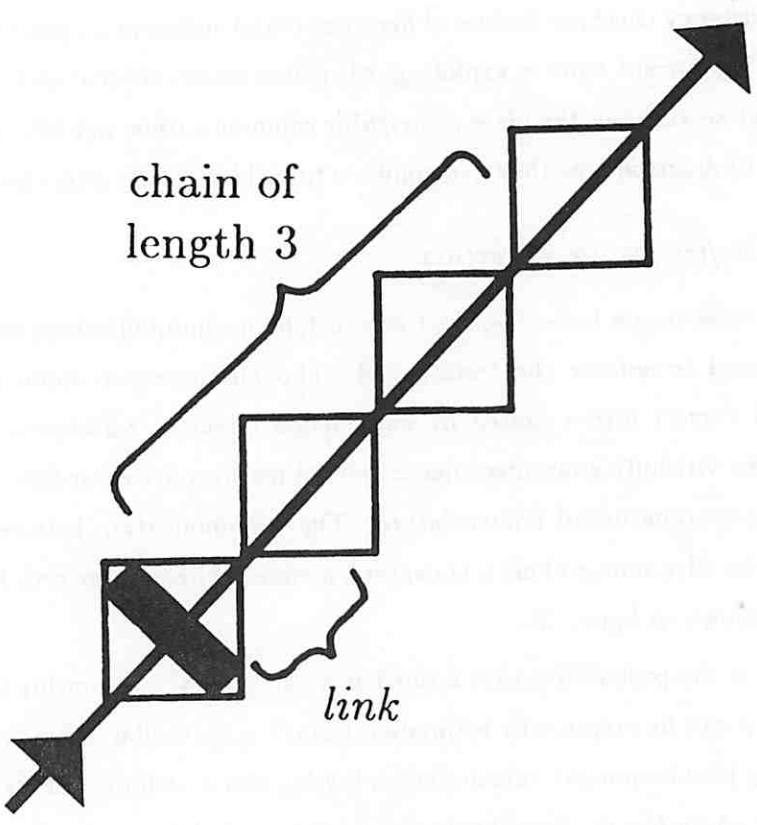


Figure 15: *Chains* of *links* along the edge trajectory.

expectations. This is directly related to the concept of chain length. When a restrictive communication scheme is used, there is a greater probability that structurally meaningful expectations will fail to propagate along the traversal trajectory; this tends to fragment structurally meaningful chains which makes them less distinguishable from chains caused by unmodelled image variations. Conversely, a less restrictive communication scheme (such as notifying a group of neighbors) virtually ensures that modelled edges will form unbroken chains along the trajectory. Less restrictive neighbor communication increases the probability (p) that a link is due to nonstructural causes, but this can be corrected by a commensurate increase in the minimal feature chain length.

D. Measuring temporal persistence

The previous sections have described how *consistent edge motion* and virtually noise-free features may be extracted. Using this framework, this section describes how *temporal persistence* may be determined using neighbor communication along the trajectory.

Remember that *temporal persistence* was defined as the period of time over which a spatial feature has stably persisted in the image plane. This interval may be determined in several ways.

One way to compute temporal persistence assumes that each edge/motion unit has an independent counter to measure elapsed time. When an edge is locally detected and no upstream neighbor has indicated its approach (i.e., the edge is newly appearing), or when the state of the edge/motion unit is reset, the local counter for that edge/motion unit can be reset to 0. This counter measures elapsed time while the edge is traversing it. As the edge moves along the trajectory, the counter value at the current edge position is then used to reset the counter of the downstream neighbor. Thusly, the total elapsed time since the edge first appeared is maintained by the edge/motion units along the edge trajectory. Though this technique is conceptually intuitive, it requires a local counter and external clock, and the maximum counter value places a limit on the ability to measure temporal persistence. These effects can be partially offset by more coarsely quantizing time and careful design of local counters.

A similar approach assigns a time stamp to a spatial feature (e.g., an edge) when it first appears on the image plane. That time stamp may then be propagated (without modification) to downstream neighbors. This approach eliminates the need for local counters, though it requires a final computation to determine the temporal persistence interval; that is, for a time stamp $\bar{t}_i$ and current time t, the current temporal persistence interval is then $t \quad \bar{t}_i$.

Temporal persistence was defined as the period of *time* over which a spatial feature has persisted. Alternatively, it may be approximated by measuring the traversed distance over the *space* of the

image plane. That is, chain length can be used to approximate the temporal persistence interval. Chain length is related to the temporal persistence interval by the real velocity at which the feature traverses the image plane. For example, within the same time interval, a slowly moving edge will have a smaller chain length than a faster moving edge. Because immediately adjacent features on a single object tend to have similar velocities, the relative chain lengths among adjacent features tends to be the same as the relative temporal persistence intervals for those adjacent features. Conversely, relative chain lengths among features moving at different velocities will differ from the relative temporal persistence intervals for those features.

As usual, the extent to which approximations may be used depends upon the way in which the information will be used. For the purpose of extracting virtually noise-free moving edges and roughly determining their persistence interval, the chain length approximation was sufficient. Alternatively, other applications (e.g., object segmentation based upon the temporal persistence interval) may justify more exact computation.

E. Time-based *versus* frame-based *image acquisition*

So far, it has been implicitly assumed that a *time-based* image acquisition paradigm is used. Conversely, most current image acquisition hardware was developed for television using a *frame-based* paradigm. Though imaging devices specially designed for computer vision may be more appropriate [34], the prevalence of frame-based devices justifies its discussion. This subsection discusses enhancements useful for frame-based imagery. It is important to note that omitting these enhancements will not result in incorrect or noisy extraction of edges. Rather, discontinuously moving edges caused by frame-based imagery will be eliminated by the time-based *context-dependent model enforcement* presented in previous subsections. These enhancements are only intended to increase the number of features extracted from frame-based imagery.

As mentioned in section IIA, frame-based techniques allow a feature in one frame to move, in principle, anywhere in the continuous space of the next frame; this introduces the *correspondence problem.* Conversely, time-based models do not discretely sample in time, hence, they need not "lose" track of image features since there is no gap of time in which the position of a feature cannot be monitored. As a result, each of the steps shown in figure 12 is affected by the use of frame-based imagery.

Temporal edges (i.e., *events*) identify portions of a locally straight moving edge. As discussed in section IIIB, frame-based imagery usually suffers from some temporal aliasing; this can introduce temporal edges that are not structurally related to moving edges. Appropriate temporal low-pass

filtering can largely eliminate the effects of temporal aliasing.

The techniques presented here and in [19] relied upon a time-based model in which a translating edge that does not disappear from the image is constrained to move through adjacent image positions. Because imagery is sampled continuously in time, a feature is detected when it traverses an event detector. Frame-based image acquisition introduces the potential problem that an event at a pixel may not occur during a temporal sampling period, hence, it may not be detected. This form of temporal aliasing undermines the determination of edge direction based upon *order of passage* and the propagation of edge information along the trajectory. For example, a moving edge might create the *order of passage* ABC, but if detector B was not traversed during a temporal sampling period, the detected *order of passage* would reduce to AC; thus, failure to detect an edge while it traverses detector B results in an incorrect *order of passage*.

The failure to detect *events* when they occur (due to frame-based image acquisition) can be partially offset by modifying the *triad* finite automaton (shown in figure 10b). Specifically, the effect of discrete temporal sampling which is inherent to frame-based imagery may be incorporated into the automaton. For example, any *order of passage* which is consistent with an edge with direction $\alpha = \pi/8$ can be incorporated into the automaton (e.g., if brackets indicate simultaneous traversal of cells: $[AC][BD]$, $C[AD]B$, $[ACD]B$, etc.). Put another way, frame-based imagery introduces additional possible arcs in the *triad* automaton. As such, additional arcs may be added to partially offset the effect of frame-based imagery.

Though the *triad* finite automaton may be modified to deal with small discontinuities caused by frame-based imagery, larger discontinuities cannot be overcome in this way. Time-based imagery allowed communication among *adjacent* neighbors along the trajectory. Conversely, frame-based imagery can result in discontinuous detection of an edge along the trajectory; hence, restricting communication to immediate neighbors can fragment a *chain* along the trajectory. To deal with this, a larger neighborhood can be used to communicate expectations across discontinuities along the edge trajectory.

Figure 16 shows a moving edge. Because frame-based imagery allows for discontinuous detection of the moving edge along the trajectory, current edge information must be communicated to all downstream neighbors that can next detect the moving edge. For example, the moving edge shown in figure 16 could communicate its edge information to all units that lay along the trajectory within a fixed radius.

There is a fixed relationship between the maximum linking radius required for frame-based imagery and the maximum allowable velocity at which a detectable feature may move across the

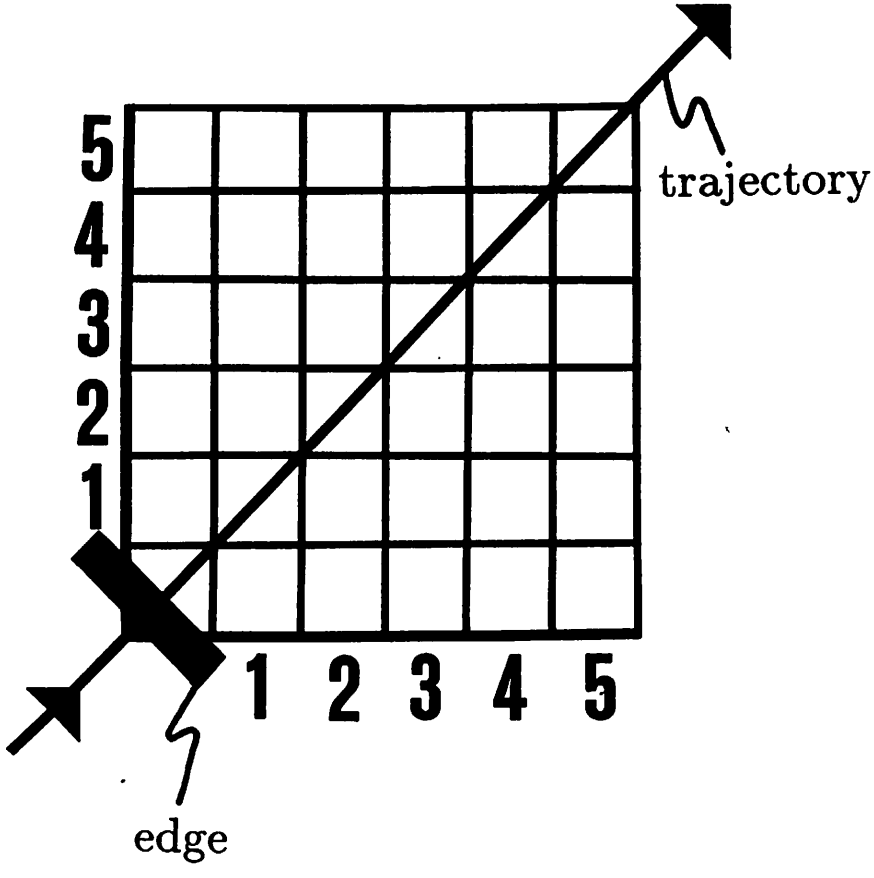


Figure 16: Increasing fixed communication radius for *frame-based* imagery.

image. Using a 256 × 256 image sampled at 30 frames per second as an example, features that can smoothly translate across the bounded image in no more than two seconds require a maximum linking radius of about 4. Doubling the maximum detectable feature velocity doubles the size of the linking radius. Thus, an effective way to deal with discontinuities created by frame-based imagery is to communicate to a larger neighborhood based upon the maximum detectable image velocity. Features moving faster than this maximum detectable velocity break *links* along the trajectory, and hence, they fail to create sizeable *chains* along the trajectory; thus, they virtually never introduce errors when a minimum chain length is enforced.

Several other factors affect the size of the linking radius. A larger linking radius increases the chance that nonstructural *links* will occur. As shown in subsection C, a greater probability of nonstructural links p can be offset by increasing the minimum chain length n. Increasing the neighborhood communication radius increases the number of communications lines; thus, physical realizability places an upper limit on the radius.

Communication within the neighborhood radius need not be indiscriminate. Instead, only a subset of neighbors within a fixed radius may be *linked* based upon local edge characteristics (e.g., velocity, local *event* density, etc.). This reduces the probability of nonstructural links (i.e., reduces p), hence, shorter minimum chain lengths would be required.

V. Results on a Natural Image Sequence

Previous sections discussed the determination of temporal persistence and consistent edge direction/motion. This section demonstrates the approach on an office scene sequence provided to the author by R.C. Bolles; the imagery was originally described by Bolles and Baker in [4]. In this dense sequence (no more than about one pixel motion per frame), the camera translated orthogonally to the optical axis (as when one looks directly out of the side window of a moving car).

Figure 17 shows a frame in the sequence before any processing has been done. Some spatial aliasing exists in the imagery because explicit low-pass spatial filtering was not done prior to image acquisition (as discussed in section IIIA). Because there was only a small amount of aliasing and the preservation of image sharpness was considered important, no steps were taken to alleviate spatial aliasing (i.e., each frame was not put through a spatial LPF).

Figure 18 shows the result of temporally filtering the original frame shown in figure 17. The temporal signal (i.e., pixel intensity over time) was convolved with a normalized half gaussian with $\sigma = 2$ to temporally filter the imagery. Notice the motion blurring caused by temporal filtering (as discussed in section IIIB).

Figure 17: Office scene image (courtesy of R.C. Bolles).

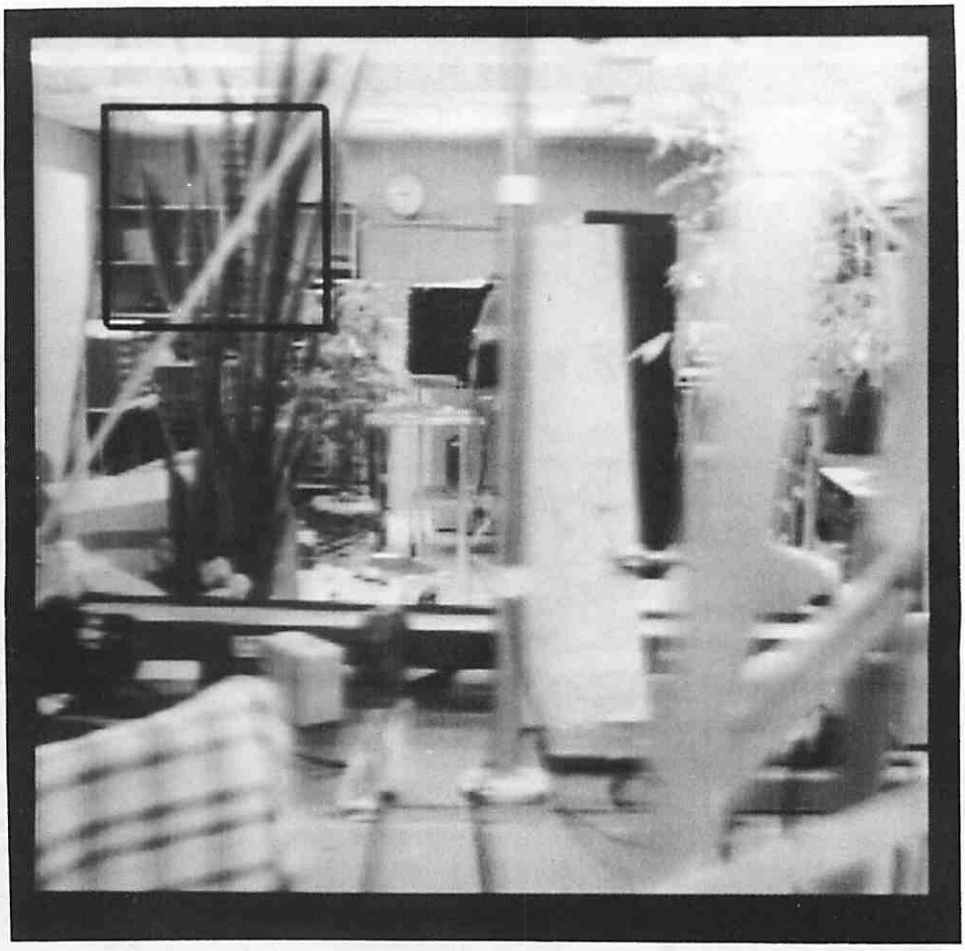


Figure 18: Office scene image after low-pass temporal filtering.

In order to develop an intuition for the motion in this sequence, figure 19 shows the motion over sixteen frames which occured in the small patch marked by a box in the upper left-hand corner of figure 18.

Figure 20 shows the temporal edges (*events*) detected in the image patch during the sixth frame; pixels at which an event has been detected appear as black spots. Temporal edges were arbitrarily defined as maxima in the temporal signal. Note the correspondence between the temporal maxima and the spatial maxima which occur in figure 19. That is, the slanted pole, the leaves on the plant, and the wires hanging from the ceiling contain spatial maxima that were detected using temporal maxima. Events detected in figure 20 deemed consistent with previous image motion are marked with a vector indicating the direction of consistent motion.

Figure 21 shows the *consistent edge motion* computed from the local image patch; the placement of edges is due to the use of temporal maxima as image events. In previous sections, it was stated that spatial structures which persist in a stable manner over time are more likely to be structurally related to objects. This can be seen by the absence of false positives in figure 21. That is, no edges are detected where an environmental structure does not occur; *all* the moving edges extracted from the sequence correspond to significant environmental structures. As discussed in section IIIC, greater edge density can be obtained by using multiple definitions of temporal events.

Rather than exactly compute the temporal persistence interval, the chain length approximation was used (as discussed in section IVD). Figure 22 shows chain lengths after fourteen frames; chain length is encoded as intensity, thus, longer chains are brighter than shorter chains. Chain lengths from previous time frames have been retained in order to show that chain length increases as consistently moving contours traverse edge/motion units along its trajectory. That is, temporal persistence increases over time for a consistently moving contour. Note that even though the wires do not markedly contrast with the background, portions of wire tend to persist over time (see upper left in figures 21 and 22).

VI. Discussion

A time-based computational model has been presented for determining temporal persistence and consistent edge motion. Previous work reported in [19] was expanded upon in order to demonstrate how edge motion may be determined from natural imagery. Propagation of information along the edge trajectory was found capable of virtually noisefree determination of temporal persistence and consistent edge motion. The overall technique was demonstrated on natural imagery.

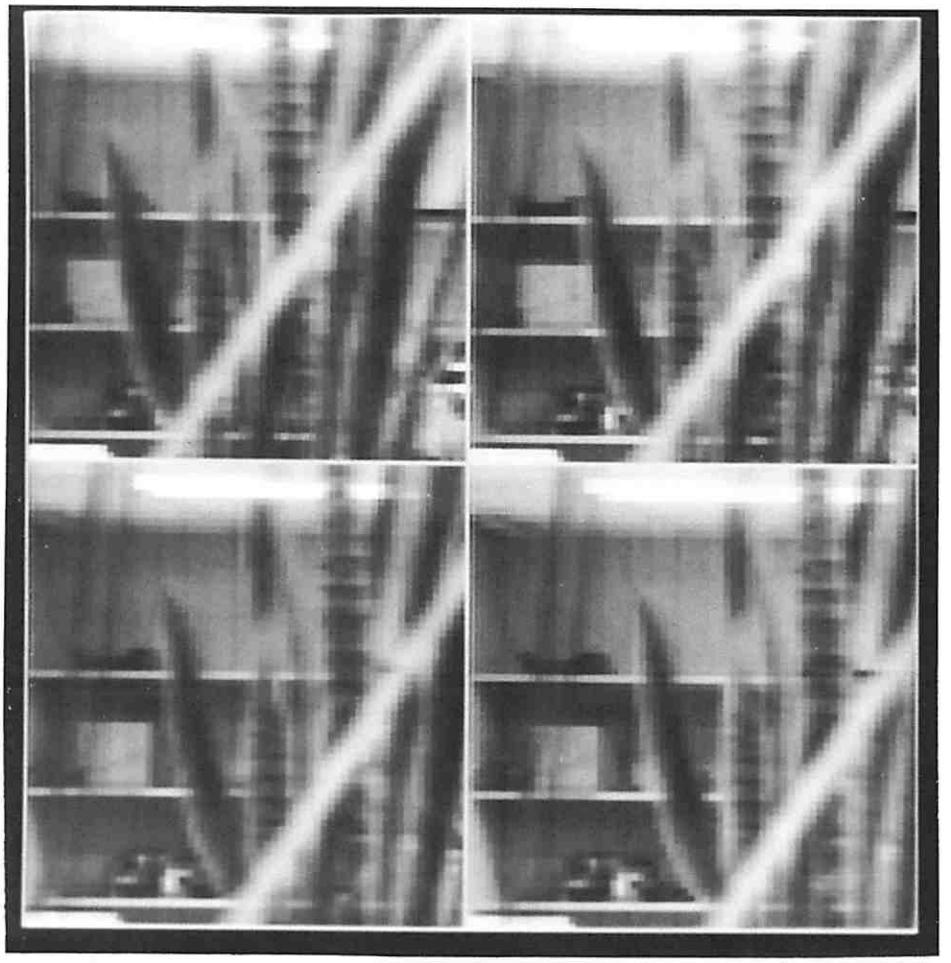


Figure 19: Motion over sixteen frames in the small patch marked by a box in the upper left-hand corner of figure 18.

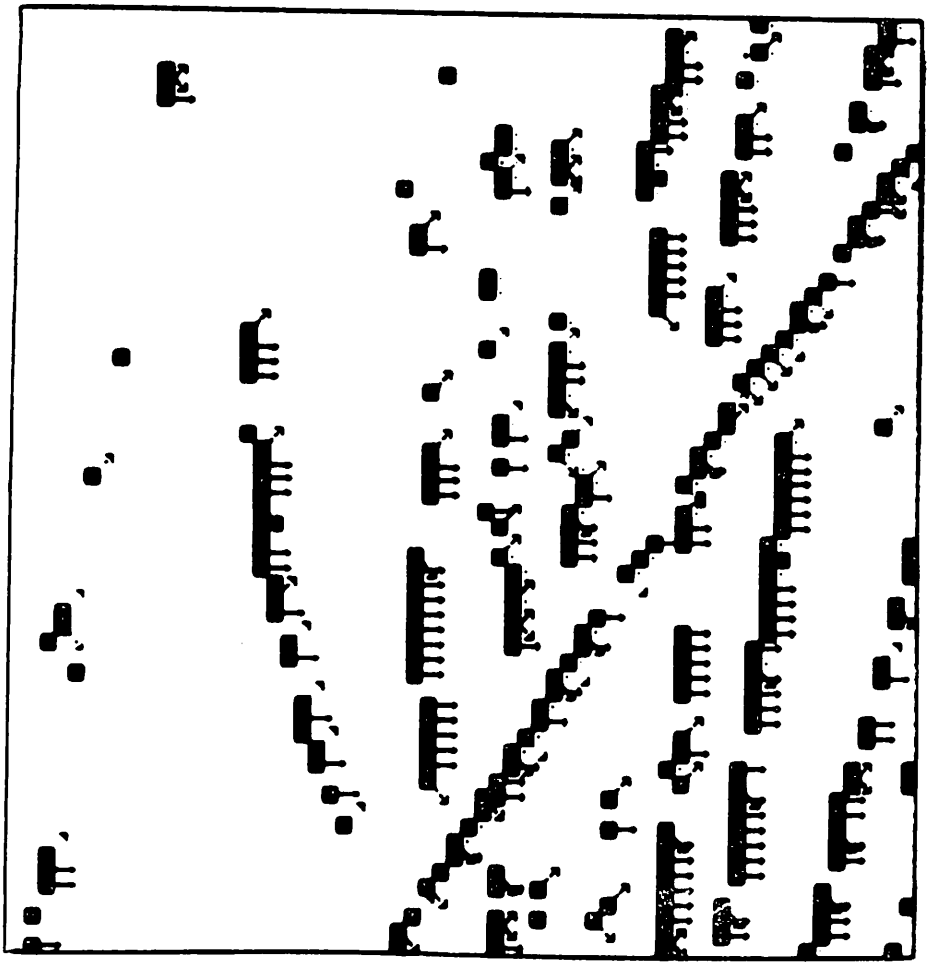


Figure 20: Temporal edges (*events*) detected in the image patch shown in figure 19 during the sixth frame; pixels at which an event has been detected appear as black spots.

Figure 21: *Consistent edge motion* computed from the local image patch.

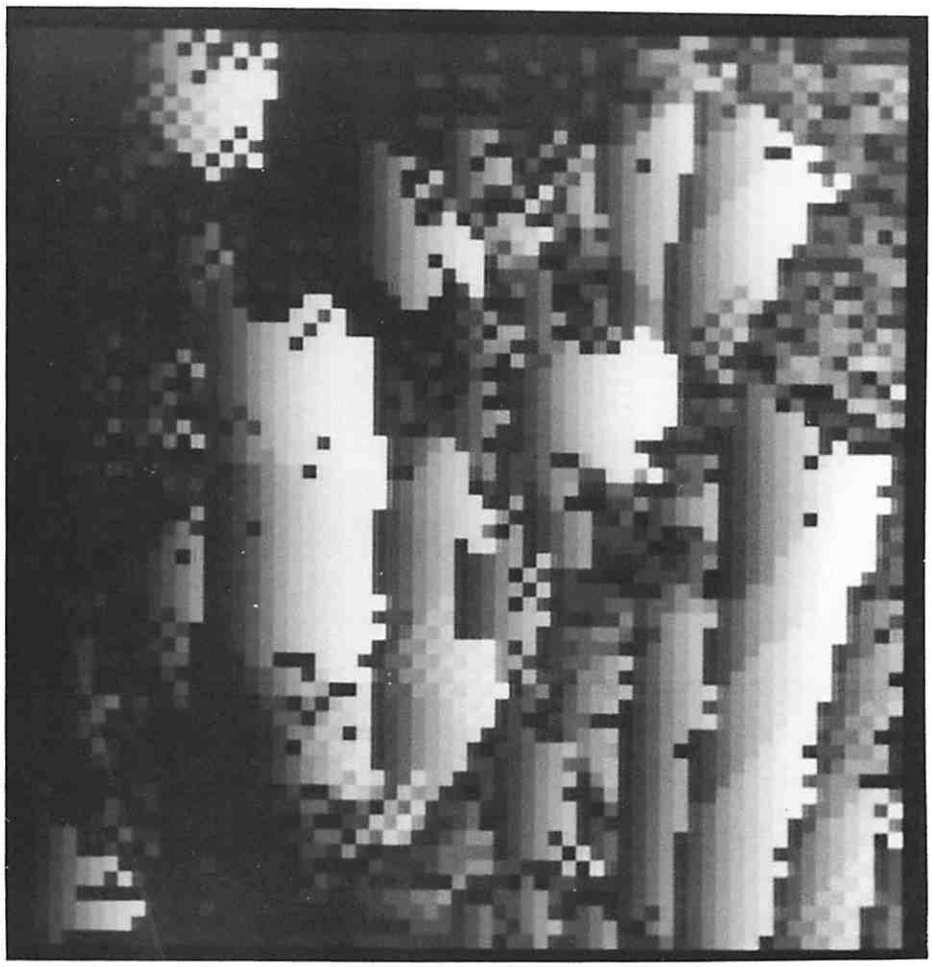


Figure 22: Temporal persistence approximated by chain length that was computed from the local image patch; chain length is encoded as intensity.

In computer vision, most object recognition work has mapped spatial characteristics of the image to idealized internal models [7,24,41]. Because spatial structures which persist in a stable manner over time are more likely to be structurally related to objects, it follows that the temporal characteristics of spatial features developed in this paper can assist in the object recognition process. These and other temporal features are important to the object recognition task [10,15,16,30]; further study is thus warranted.

Current image acquisition and processing hardware was designed for purposes other than computer vision (e.g., television, sequential program execution). As discussed in this paper and in [15,34], this conventional hardware is not well-suited for the task of acquiring and processing natural imagery. Though the model developed in this paper can work within the constraints of currently available hardware, it is especially well-suited for fully parallel implementation. In particular, the processing elements are uniform throughout the image plane, only local adjacent communication is required, and the processing may be simply implemented using standard digital logic or analog circuits. VLSI techniques can be used to create devices far better suited for the task of computer vision [34].

Because there are common vision problems faced by both man-made computers and biological organisms, one would expect there to be similarities between their solutions. This paper has developed a model for computer vision, though there is evidence that this time-based model has biological correlates. As previously discussed, the event-driven edge/motion technique developed in [19] and extended in this paper is most related to the opponent correlation models which have been advanced primarily as models of biological motion processing [1,29,32,33,38,39]. Other biological correlates are given in [19]. In addition, the use of past motion measurements to constrain current motion interpretation has been found to occur in natural vision systems [35]. Though not conclusive, it is plausible that a form of the model presented in this paper underlies a description of biological visual processing.

Most of the unresolved issues cited in [19] have been addressed by this paper. As is usually the case, new issues have been identified. Since temporal persistence and consistent edge motion are newly defined features, a fuller understanding of their potential use for image understanding is needed. Further, it appears that other temporal features may be extracted in an analogous fashion. For example, optical flow may be locally determined at very sparse points in the image by noting that the aperture problem does not occur at places where contours terminate; since most line segments terminate in the bounded space of the image, sparse real velocities may be simply determined by tracking the termination points of moving line segments. Current work is

also developing a fuller understanding of methods for propagating information along the trajectory, determining local consistency, and measuring temporal persistence. Future work will examine the effect and use of temporal features for perceptual organization and object recognition. It is hoped that this approach to computer vision and motion analysis will provide a much better understanding of the complexities of our visual world.

Acknowledgements

I thank P. Anandan for his acute observations, expertise, and sharing of mutual interests. I am also endebted to P. Balasubramanyam, D. Strahman, M. Snyder, I. Pavlin, J.B. Burns, and J. Dolan for their helpful comments and suggestions. The data kindly provided by R.C. Bolles and H.H. Baker was of great help.

References

[1] E.H. Adelson and J.R. Bergen, "Spatiotemporal Energy Model for the Perception of Motion," *Journal of the Optical Society of America: A*, vol. 2, no. 2, pp. 284-99, February 1985.

[2] J.A. Allocca and A. Stuart, *Transducers: Theory and Application*, Reston Publishing: Reston, VA, 1984.

[3] P. Anandan, "Computing Optical Flow from Two Frames of a Dynamic Image Sequence," COINS Technical Report 86-16, Department of Computer & Information Science, University of Massachusetts at Amherst, 1986.

[4] R.C. Bolles and H.H. Baker, "Epipolar-Plane Image Analysis: A Technique for Analyzing Motion Sequences," *Proceedings of the DARPA Image Understanding Workshop*, pp. 137-148, Dec. 1985.

[5] S. Bottini, "On the Visual Motion Blur Restoration," *2nd IEEE International Conference on Visual Psychophysics and Medical Imaging*, Brussels, pp. 143–149, 1981.

[6] S. Bharwani, E.M. Riseman, and A. Hanson, "Refinement of Environmental Depth Maps Over Multiple Frames," *Proceedings of the Workshop on Motion: Representation and Analysis*, Charleston, SC, pp. 73–80, May 1986.

[7] J.B. Burns and L.J. Kitchen, "Recognition in 2D Images of 3D Objects from Large Model Bases Using Prediction Hierarchies," *10th International Joint Conference on Artifical Intelligence*, Milan, Italy, August 1987 (to be published).

[8] P.J. Burt, "Fast Filter Transforms for Image Processing," *Computer Graphics and Image Processing*, vol. 16, pp. 20-51, 1981.

[9] P.J. Burt, "Stimulus Organizing Processes in Stereopsis and Motion Perception," Ph.D. dissertation, Department of Computer & Information Science, University of Massachusetts at Amherst, June 1976.

[10] J.E. Cutting, *Perception with an Eye for Motion*, MIT Press: Cambridge, MA, 1986.

[11] Duda and Hart, *Pattern Classification and Scene Analysis*, Wiley: NY, 1973.

[12] D.J. Fleet and A.D. Jepson, "Spatiotemporal Inseparability in Early Vision: Centre-Surround Models and Velocity Selectivity," *Comput. Intell.*, vol. 1, pp. 89–102, 1985.

[13] L. Frishman, "The Velocity-Tuning of Neurons in the Lateral Geniculate Nucleus and Retina of the Cat," Ph.D. dissertation, Department of Psychology, University of Pittsburgh, Pittsburgh, PA, 1979.

[14] D. Gabor, "Theory of Communication," *Journal of the Institute of Electr. Eng.*, vol. 93, pp. 429-57, 1946.

[15] J.J. Gibson, *The Ecological Approach to Visual Perception*, Houghton-Mifflin: Boston, MA, 1979.

[16] E.B. Goldstein, *Sensation and Perception*, Wadsworth Publishing: Belmont, CA, 1980.

[17] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," M.I.T. A.I. Memo 572, April 1980. (See also, Artificial Intelligence, vol. 17, nos. 1-3, pp. 185-203, August 1981.)

[18] P. Kahn, "Event-Driven, Context-Dependent Feature Extraction from Time-Varying Imagery: Research Review," Computer Vision Research Laboratory working paper, University of Massachusetts at Amherst, Amherst, MA, Nov. 1986.

[19] P. Kahn, "Local Determination of a Moving Contrast Edge," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. PAMI-7, no. 4, pp. 402-409, July 1985.

[20] J. Krauskopf, "Effect of Retinal Stabilization on the Appearance of Heterochromatic Targets," *Journal of the Optical Society of America,* vol. 53, pp. 741-743, June 1963.

[21] L.L. Lapin, *Statistics: Meaning and Method,* Harcourt Brace Jovanovich: NY, 1975.

[22] B.P. Lathi, *Communications Systems,* Wiley: NY, 1968.

[23] D. Lawton, "Processing Dynamic Image Sequences from a Moving Sensor," COINS Technical Report 84-05, Department of Computer & Information Science, University of Massachusetts at Amherst, 1984.

[24] D.G. Lowe, *Perceptual Organization and Visual Recognition,* Kluwer Academic Publishing, 1985.

[25] D.C. Marr and E.C. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London: B,* vol. 207, pp. 187-217, February 1980.

[26] C.D. McGillem and G.R. Cooper, *Continuous and Discrete Signal and System Analysis,* 2nd edition, Holt, Rinehart & Winston: NY, 1984.

[27] J.M. Prager, "Segmentation of Static and Dynamic Scenes," COINS Technical Report 79-7, Department of Computer & Information Science, University of Massachusetts at Amherst, 1979.

[28] H. Shariat and K.E. Price, "How to Use More Than Two Frames to Estimate Motion," pp. 119–124, *Proceedings of the Workshop on Motion: Representation and Analysis,* Charleston, SC, May 1986.

[29] W. Reichardt, "Autokorrelationsauswertung als Funktionsprinzip des Zentralnervensystems," *Z. Naturforsh,* Teil B12, pp. 447-457, 1959. (See also, "Autocorrelation, a Principle for the Evaluation of Sensory Information by the Central Nervous System," in *Sensory Communication,* W.A. Rosenblith, ed., Wiley: NY, 1961.)

[30] I. Rock, *The Logic of Perception,* MIT Press: Cambridge, MA, 1983.

[31] M.S. Rodin, *Digital and Data Communication Systems,* Prentice-Hall: Englewood Cliffs, N.J. 1982.

[32] J.P.H. van Santen and G. Sperling, "Elaborated Reichardt Detectors," *Journal of the Optical Society of America: A,* vol. 2, no. 2, pp. 300-21, February 1985.

[33] J.P.H. van Santen and G. Sperling, "Temporal Covariance Model of Human Motion Perception," *Journal of the Optical Society of America: A,* vol. 1, no. 5, pp. 451–473, May 1984.

[34] M.A. Silvotti, M.A. Mahowald, and C.A. Mead, "Real-Time Visual Computations Using Analog CMOS Processing Arrays," *Advanced Research in VLSI, Proceedings of the 1987 Stanford Conference,* P. Losleben, ed., MIT Press: Cambridge, MA, pp. 295–312, 1987.

[35] M.V. Srinivasan, S.B. Laughlin, and A. Dubs, "Predictive Coding: a Fresh View of Inhibition in the Retina," *Proceedings of the Royal Society of London: B,* vol. 216, pp. 427–459, 1982.

[36] C.W. Tyler, "Analysis of Visual Modulation Sensitivity. II. Peripheral Retina and the Role of Photoreceptor Dimensions," *Journal of the Optical Society of America: A,* vol. 2, no. 3, pp. 393–398, March 1985.

[37] S. Ullman, "Analysis of Visual Motion by Biological and Computer Systems," *IEEE Transactions on Computing,* vol. C-14, pp. 57-69, Aug. 1981.

[38] A.B. Watson and A.J. Ahumada, "Model of Human Visual-Motion Sensing," *Journal of the Optical Society of America: A,* vol. 2, no. 2, pp. 322-42, February 1985.

[39] A.B. Watson and A.J. Ahumada, "A Look at Motion in the Frequency Domain," NASA Technical Memo. 84352, 1983.

[40] A.M. Waxman and K. Wohn, "Image Flow Theory: A Framework for 3-D Inference from Time-Varying Imagery," *Advances in Computer Vision,* C. Brown, ed., Erlbaum Publishers (to be published).

[41] T.E. Weymouth, "Using Object Descriptions in a Schema Network for Machine Vision," Ph.D. dissertation, University of Massachusetts, Amherst, MA, 1986.

[42] M. Young, *Optics and Lasers: An Engineering Physics Approach,* Springer-Verlag: NY, 1977.