Asynchrony in image analysis: using the luminance-to-response-latency relationship to improve segmentation

Pierre-Yves Burgi

The Smith-Kettlewell Eye Research Institute, 2232 Webpster Street, San Francisco, California 94115

Thierry Pun

Artificial Intelligence and Vision Group, Computing Science Center, University of Geneva, 24 rue Général Dufour, 1211 Geneva 4, Switzerland

Received August 9, 1993; accepted Jannary 3, 1994

We deal with the problem of segmenting static images, a procednre known to be difficnlt in the case of very noisy patterns. The proposed approach rests on the transformation of a static image into a data flow in which the first image points to be processed are the brighter ones. This solution, inspired by human perception, in which strong luminances elicit reactions from the visual system before weaker ones, has led to the notion of asynchronous processing. The asynchronous processing of image points has required the design of a specific architectnre that exploits time differences in the processing of information. The results obtained when very noisy images are segmented demonstrate the strengths of this architecture; they also suggest extensions of the approach to other computer vision problems.

Key words: segmentatiou, asynchronous processing, nonlinear diffnsion, visnal data flow, latency, human visual system.

1. INTRODUCTION

Segmentation plays an important role in a vision system, as it permits the identification of regions characterized by a specific luminance, color, or texture.[1-4] Although various image features could be combined to help in the task of segmentation, in this paper we focus on finding regions by using only the luminance information. This problem is not new, as many authors have proposed techniques (for example, based on nonlinear diffusion) for extracting regions.[5-7] The novelty in this paper is the demonstration that classical techniques of segmentation can be significantly improved if the luminance information is seen and processed as a temporal data flow.

How can luminance information be related to temporal delays? The answer is based on the notion of the luminance-to-response-latency relationship, which has been inspired by the human visual system.[8] For decades it has been well known to psychologists that there exists a dependence of reaction time on stimulus intensity, in which stronger signals elicit responses before weaker ones.[9] Visual illustrations of this phenomenon are the Pulfrich effect and the Hess effect. The Pulfrich effect[10] described in 1922 by Carl Pulfrich, is easily experienced by an observer looking at a swinging pendulum bob while wearing two different lenses, one being dark and the other clear: the path of the bob swinging in the frontoparallel plane of the observer appears approximately elliptical in depth. This stereoscopic visual illusion results from an increase in processing time in one eye (the one looking through the dark lens), which, as has recently been proposed, could directly affect the distribution of the activity of disparity-selective cortical cells in the visual cortex.[11] In the Hess effect,[12] first reported by Carl von Hess in 1904, an observer looks monocularly at two targets that are moving laterally at the same velocity (also in the observer's frontoparallel plane). When one target is reduced in luminance, the corresponding increase in visual latency causes a change in their relative apparent locations of the targets: the observer reports one target as though it were trailing the other.

Although in this paper we concentrate on using the luminance-to-response-latency relationship, it must be emphasized that visual latencies in the human visual system are also dependent on other visual features (e.g., color,[13] spatial frequency,[14] and contrast[15]). The paper is organized as follows: Section 2 addresses the general problem of processing information that is distributed in time. Section 3 presents our model of asynchronous segmentation. It is followed in Section 4 by experimental results that compare, for the same architecture, the results of segmentation obtained with and without a luminance-to-time relationship. Other possible applications of our model are discussed in Section 5. Finally, Section 6 presents some concluding remarks.

2. PROCESSING A DATA FLOW

From the point of view of classical image processing,[16] a static image is considered to be an array of image points that must be processed as a whole, or synchronously. Conversely, the introduction of a relationship between image variables and time leads to the notion of asynchrony. Asynchrony indicates that image points are not processed as a whole but instead as a temporal sequence

0740-3232/94/061720-07$06.00

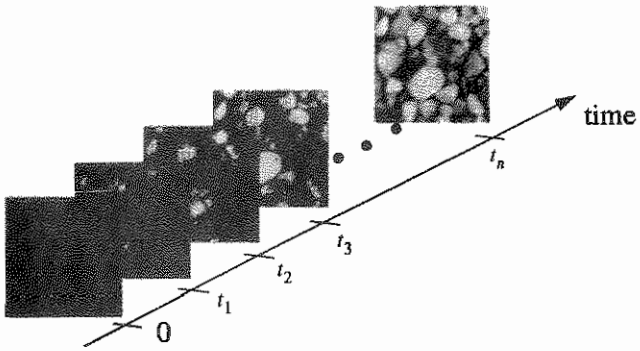

Fig. 1. Illustration of the effect of the luminance-to-response-latency relationship. At time 0 no image points are considered. As time increases, more and more image points are considered, those having a high luminance being taken into account first.

or data flow. For a static image to be transformed into a data flow, the latency or delay associated with an image variable (luminance in our case) must be determined for every image point, or pixel. (In a more general manner, a delay might be associated with every primitive, such as edge element or region.) This transformation is schematically illustrated in Fig. 1. In the figure, it is shown that the introduction of latencies creates a temporal sequence in which the most-luminous image points can activate the processing stages before the darker ones can.

Usually, for the human visual system the relationship of luminance to response latency is reported as being nonlinear and possibly approximated by an inverse cube-root function.[17] This function decreases rapidly at low luminances and slowly at high luminances. So far, this nonlinearity has not been found to be relevant for our model, and for that reason a linear approximation was chosen. For an image containing a range of luminances, the luminance-to-response-latency relationship yields a data flow that must be processed continuously. Practically, a sampling time is defined, and, consequently, the data flow is discretized into a temporal sequence of image points.

How could segmentation benefit from asynchrony? It has been demonstrated elsewhere by means of the concept of mutual information[18] that the general problem of classifying noisy symbols that are characterized by specific amplitudes (a problem similar to segmentation) is naturally well suited to asynchronous processing. In particular, there is a period of time during which the available information, corresponding to the strongest amplitudes, has not yet been corrupted by noise of correspondingly lower amplitude. During this period of time the probability of misclassifying the information is minimized. Because we deal with an unsupervised classification, one of the main problems is to know when this optimum period occurs. It is at this point that we wish to design an architecture that uses this optimum period without having to know it in advance. Also, we propose two strategies that are based on temporal interactions between image points previously processed and those subsequently processed. The first and simplest strategy that would provide such interaction is to use a temporal integration at the output of the processing stages, as illustrated in Fig. 2(a). This allows the solution appearing during the optimum period to contribute to the final result. A second strategy that would allow an interaction between the current image information content and the forthcoming one is to use feedback in the processing stages, as shown in Fig. 2(b). A combination of temporal integration and feedback has been adopted in our model, leading to the architecture presented in the next section.

3. MODEL OF ASYNCHRONOUS SEGMENTATION

The segmentation of an image that has been transformed into a data flow is said to be asynchronous. One of the main features of asynchronous segmentation is its temporal dimension, which somehow complicates the structure of the different stages of processing. To make the understanding of our model easier, we have preceded the description of the equations by a general overview that qualitatively presents the mechanism of asynchronous segmentation.

A. General Overview

Segmentation can be subdivided into two complementary tasks: (1) the detection of edges and (2) the extraction of regions delimited by edges. One possible strategy is thus to detect the high spatial frequencies contained in boundaries and then smooth the low spatial frequencies of homogeneous regions inside these boundaries. This scheme implies that high and low spatial frequencies must be preserved. In our model we use a rotationally symmetrical (so-called isotropic) filter that enhances edges, whatever the orientation, and preserves uniform regions.

In accordance with the strategies described in Section 2, this filter is applied to a data flow that corresponds to image points of a given luminance and latency, and the result is temporally integrated (Fig. 3). A particular feature of such a temporal approach is that spatially connected image points in the original image may become disconnected at a certain time if their respective latencies differ. The output of the temporal integration is thus smoothed out so that isolated spots are spread into regions, this smoothing being done through

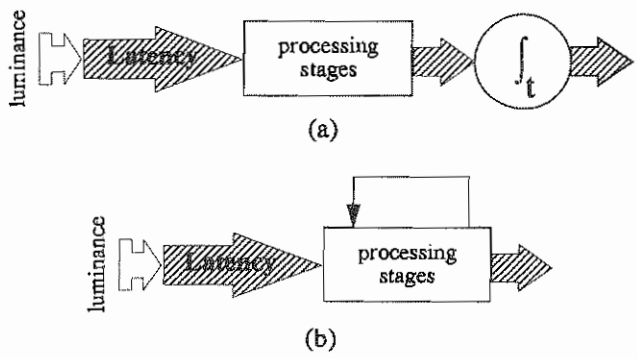

Fig. 2. Illustration of two strategies that allow early image points to interact with later ones. The hatched arrows indicate that a temporal component has been added to the spatial components, forming a data flow by associating latencies, or delays, with luminance values. (a) A temporal integration at the output of the processing stages memorizes the temporal evolution; (b) feedback permits a dynamic interaction between a current state in the processing stages and incoming image points.

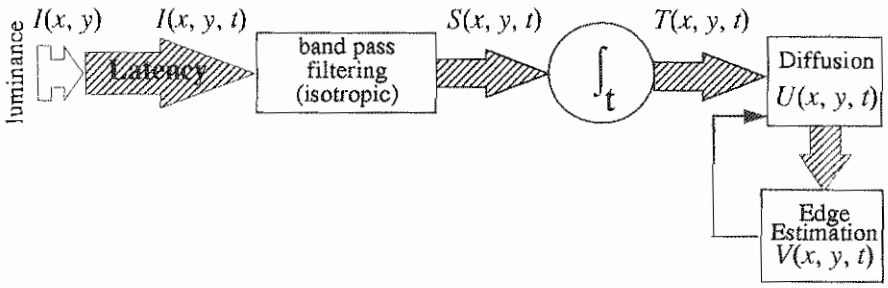

Fig. 3. Model of asynchronous segmentation. The first stage applies a bandpass isotropic filter to the data flow $I(x,y,t)$. The result $S(x,y,t)$ is temporally integrated, and the output $T(x,y,t)$ is fed to the diffusiou stage $[U(x,y,t)]$, which is dynamically controlled by the edge estimation $V(x,y,t)$.

a diffusion mechanism. If nothing stops the diffusion, the final result will be a uniform region covering all the image. Edges that have been enhanced by the isotropic filtering are used as a means of controlling the diffusion.

One implication of using a data flow is that the quality of the edge location varies with time. To be convinced of this, one can imagine a noisy image containing a bright object on a dark background. Early in time very few image points can be used to estimate the edge location, leading to a poor estimate. At the other extreme, when all image points are used (time $\gg 0$), the estimate is also poor because of interference between the background and the object. Intuitively, an intermediate time exists at which estimation is based on an optimum number of image points, that is, many foreground and few background points (as already discussed in Section 2). One effect of the temporal integration is thus to keep track of the best estimate of the edge's location.

Another way of improving the estimate of edge location is to search for spots, at the level of the diffusion stage, that are locally aligned along a specific orientation, a global approach called cooperation. The new estimate of edge location is in turn used to control the diffusion, achieving a feedback, as shown in Fig. 3. Such an architecture is thus in agreement with the conditions one needs to benefit from latencies; these conditions were presented in Section 2. The remainder of this section describes in a formal way the various stages of processing used in our model.

B. Luminance-to-Response-Latency Conversion

The first stage of our model must convert and associate latencies to luminances in order to create the data flow. The range of luminances is normalized so that the maximum $\hat{I}$ of possible luminances over all images is related to the minimum latency $\tau_{\min}$ and zero luminance is related to the maximum admissible latency $\tau_{\max}$.

The following strategy was used to process the image points asynchronously: At time zero no image points are processed; at time $t = \tau_{\min}$, all image points of maximum possible intensity are processed; in a general way, at time $t = t_n$, all image points $P(x,y)$ whose intensity meets the condition $\tau_{\max} + [I(x,y)/\hat{I}](\tau_{\min} - \tau_{\max}) \leq t_n$ are being processed. To correspond to the range of latencies found in biological systems,[19] $\tau_{\min}$ and $\tau_{\max}$ have been set to 2 ms and 22 ms, respectively.

C. Isotropic Filtering and Temporal Integration

The following stage, the isotropic filtering, is the first stage that processes a data flow. The convolution $(*)$ of a spatially distributed data flow $I(x,y,t)$ with the isotropic filter is expressed by the following equations (adapted from Grossberg *et al.*[5]):

$$R(x,y,t) = \frac{[BC(x,y) - DE(x,y)] * I(x,y,t)}{A + [C(x,y) + E(x,y)] * I(x,y,t)}, \quad (1)$$

$$C(x,y) = K_1 \exp[-(x^2 + y^2)/\alpha^2], \quad (2)$$

$$E(x,y) = K_2 \exp[-(x^2 + y^2)/\beta^2], \quad (3)$$

where B and D define a positive and a negative saturation, respectively; A, K_1, and K_2 are constants, and α and β are the radii of the Gaussian distributions. To stay close to the way that the human visual system perceives light, we normalize by the denominator the difference of the two weighted Gaussians $C(x,y)$ and $E(x,y)$ in the numerator to compute a local contrast in a manner approximating that of a Weber law, known in psychophysics as the law for the perception of contrasts over a large range of luminances.[20] To ensure that $R(x,y,t)$ is different from zero for homogeneous regions, we can show that the condition $(BK_1\alpha^2)/(DK_2\beta^2) > 1$ must be verified. Because the convolution of the image with this filter is dependent on the contrast direction (black to white or white to black), the response $R(x,y,t)$ is half-wave rectified in order to keep only the contrast information. Half-wave rectification, given by $S(x,y,t) = \max[0, R(x,y,t)]$, keeps only positive responses and suppresses all negative responses. The next stage temporally integrates the output of this rectification by summing the values obtained after every time interval:

$$T(x,y,t) = \int_0^t S(x,y,t^*)\mathrm{d}t^*. \quad (4)$$

D. Diffusion

The diffusion-mechanism stage has two main controlling pieces of information: (1) inputs from the data flow and (2) control by the contour estimates. These two constraints can be better understood if we consider an analogy with the heat-diffusion equation. In the context of the heat-diffusion equation, the first constraint is equivalent to saying that a surface is locally heated at different locations, the temperature and the number of sources varying with time. The second constraint implies that the thermic conductances are not constant and can differ from one spatial orientation to another (the diffusion is thus so-called anisotropic). Such a diffusion can be described by the following equation:

$$\frac{\partial}{\partial t} U(x,y,t) = \mathrm{div}[c(x,y,t)\nabla U(x,y,t)] + T(x,y,t), \quad (5)$$

where the divergence operator div and the gradient operator ∇ are applied to the space variables, $c(x,y,t)$ is the conductance coefficient, and $U(x,y,t)$ represents the spatiotemporal evolution of the diffusing quantity. The data flow is represented in this equation by the term $T(x,y,t)$, which in our model is the result of isotropic filtering after temporal integration. Diffusing a filtered version of the original image is not new and has been described elsewhere.[5,21] The novelty is rather in the diffusion of a filtered image that is time dependent $[T(x,y,t)]$.

The conductance coefficient $c(x,y,t)$ in Eq. (5) is critical for control of the diffusion, particularly to stop smoothing where there is a contour. This coefficient is thus a function of the estimate of the location of the contours and must be small when a large spatial gradient (corresponding to an edge) is present in order for the diffusion within the edges to he constrained. A simple estimate of the contour location $[V(x,y,t)]$ could be given by the spatial gradient of $U(x,y,t)$. However, because the asynchronous approach may yield isolated spots during the early part of the simulation, a gradient estimate of contour would be poor. For that reason, the contour location is estimated in a more global way by use of oriented Gabor filters (see below). These filters tend to respond well to points aligned along their own specific orientations. The function relating the edge strength to the conductance must be carefully chosen so as to blur small discontinuities and sharpen the contours.[6] Besides the choice of this function, there is the problem of defining what is a small discontinuity. To address this problem we use a function $g(\cdot)$, defined by

$$c(x,y,t) = g[V(x,y,t)] = \frac{\delta}{1 + \epsilon V(x,y,t)}, \tag{6}$$

that blurs all discontinuities, and we use the mechanism of competition to sharpen edges. Competition is defined within a local neighborhood where the strongest edge (response of the Gabor filters) suppresses all the weaker responses. Note that the two parameters δ and ϵ are critical for the control of the blurring (in particular, its propagation rate with time).

E. Edge Estimation

Contour locations are estimated by combining the response of oriented Gabor filters. For orientation k the edge estimation $V^k(x,y,t)$ is produced by convolving the current result of diffusion with a Gabor filter $G^k(x,y)$:

$$V^k(x,y,t) = |U(x,y,t) * G^k(x,y)|, \tag{7}$$

the definition of the Gabor filter $G^k(x,y)$ being

$$\begin{aligned} G^k(x,y) = {} & \psi \exp[(x \cos k + y \sin k)^2/\zeta^2] \\ & \times \exp[(-x \sin k + y \cos k)^2/\eta^2] \\ & \times \sin[2\pi(x+y)], \end{aligned} \tag{8}$$

where ζ and η define the shape of the filter and ψ defines its amplitude. The absolute value of the right-hand term in Eq. (7) is taken in order that $V^k(x,y,t)$ be independent of the sign of contrast. A local competition is then applied along n orientations (details can be found elsewhere[18]) to keep the strongest responses $\hat{V}^k(x,y,t)$, which are finally combined to determine $V(x,y,t)$:

$$V(x,y,t) = \frac{1}{n} \sum_k \hat{V}^k(x,y,t). \tag{9}$$

F. Implementation

Discretization of the above equations did not present a major problem. The diffusion equation [Eq. (5)] can be spatially discretized on a square lattice by use of a neighborhood of the four-nearest-neighbors N:

$$\frac{\mathrm{d}}{\mathrm{d}t} U_{ij}(t) = \sum_{(p,q)\in N} [U_{pq}(t) - U_{ij}(t)] c_{pq} + T_{ij}(t), \tag{10}$$

where c_{pq} are the conductance coefficients that are related to the horizontal and vertical estimations of the contour locations (see Fig. 7 below):

$$V_{ij}^{\mathrm{hori}}(t) = \frac{1}{n} \sum_k \hat{V}_{ij}{}^k(t) \cos(k), \tag{11}$$

$$V_{ij}^{\mathrm{vert}}(t) = \frac{1}{n} \sum_k \hat{V}_{ij}{}^k(t) \sin(k). \tag{12}$$

The convolution (isotropic and Gabor filtering), diffusion, and competition are well suited to parallel implementation. Indeed, a parallel implementation of our model on a Connection Machine CM2a (8192 processors, 256-MByte memory) was shown to be very efficient.

The parameters for the different processing stages were as follows: isotropic filtering: $A = 100$, $B = 90$, $D = 60$, $K_1 = 4$, $K_2 = 1$, $\alpha = 1.2$, $\beta = 2.4$; diffusion: $\delta = 10,000$, $\epsilon = 1$; edge estimation: $\psi = 0.63$, $\zeta = 0.37$, $\eta = 0.68$; ten nearest neighbors for the competition; $n = 2$, except for processing of the image pebbles (see Fig. 9 below), where $n = 8$.

4. Experimental Results

The role of the latencies can be demonstrated experimentally when very noisy images are processed. We considered 128×128 pixel images containing an object of luminance L_1 on a background of luminance L_2 with the condition $L_2 < L_1$ and where every image point has been perturbed by additive white noise. According to our model the probability that the first image points to be processed are those pertaining to the object is higher than the probability that these image points are those pertaining to the background. This principle is illustrated in Fig. 4, in which the evolution of diffusion is shown at different times. The original image, one half set to luminance 124 (background) and the other half set to luminance 130 (object), is perturbed by additive Gaussian white noise with a signal-to-noise ratio (SNR) of 0 dB. At time 5 ms [Fig. 4(b)] the strongest responses of the diffusion are those produced by the object. The boundary separating the object from the background is already apparent. At a later time [10 ms, Fig. 4(c)], image points from the background have also been processed, but their delayed actions prevent them from spatially interfering with the object; thus a well-defined boundary results, particularly at time 20 ms [Fig. 4(d)]. This result can be compared with synchronous processing, shown in Fig. 5, in which all conditions are identical to the asynchronous processing (same architecture, same parameters, same diffusion, same sampling time for the diffusion) except for the lack of delays in the arrival of the image points. Synchronous processing results in a very noisy image, which is particularly apparent when a threshold is applied to label the figure and the background, as shown in Fig. 6. For both results

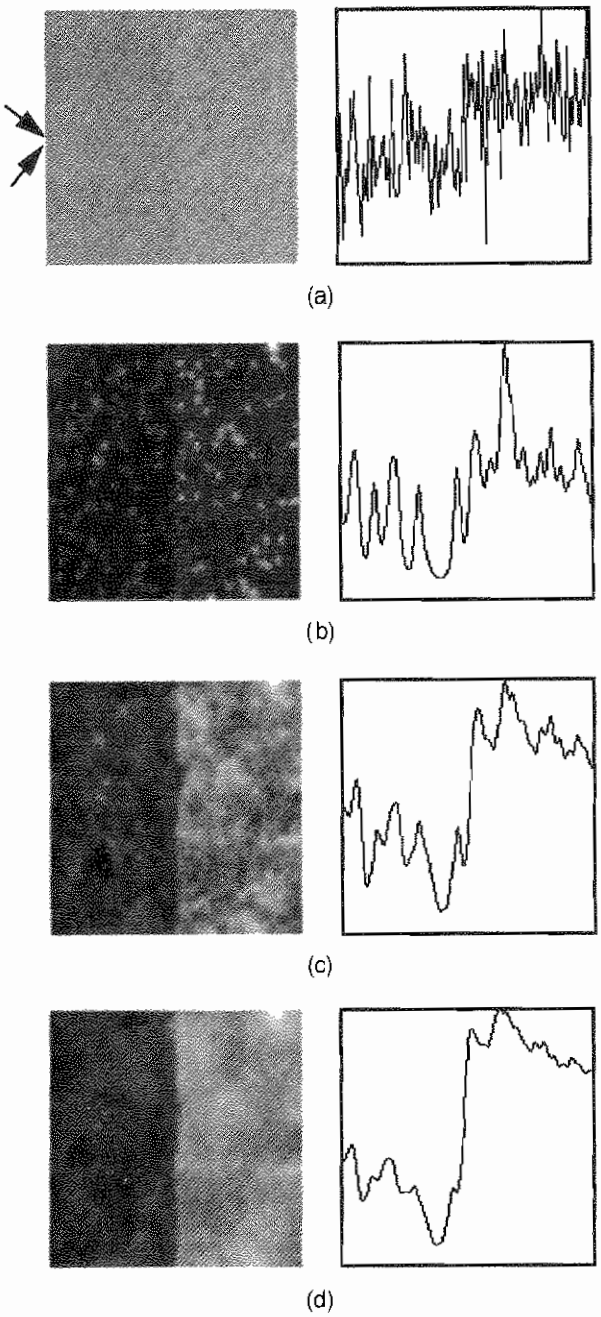

Fig. 4. Temporal evolution of asynchronons diffusion: (a) original image (128 × 128 pixels) with one half set to luminance 124 and tbe other half set to 130, both perturbed by additive Gaussian white noise with a standard deviation of 6. The SNR is 0 dB. The average of three lines, which are delimited by the two arrows in the original image, is shown at the right. The other images are the results of the diffusion (time increment 1 ms) at times (b) 5 ms, (c) 10 ms, and (d) 20 ms. This sequence of images shows how the boundary becomes less noisy as time progresses, a consequence of the continual interaction betweeu early and later image points.

(asynchronous and synchronous), the mean value has been chosen as threshold. Figure 6(a) shows the effect of thresholding the image presented in Fig. 4(d): while the border separating the figure from the background is not straight, the surfaces are noiseless. In contrast, Fig. 6(b) shows that thresholding the image presented in the left-hand panel of Fig. 5 results in noisy regions in which figure and background are not well defined.

Two questions pertaining to the dynamic nature of the process must be addressed. First, at which time must the simulation be stopped? Second, what is the influence of the choice of the sampling-time value?

To answer the first question, one must remember that at time $t = \tau_{max}$ all the visual information has been processed. Nevertheless, this situation does not prevent the simulation from continuing, because the diffusion is still active. Also the criterion that is used to determine when the simulation must be stopped is the smoothness of the surfaces resulting from the diffusion, a process known to be slower than the sharpening of edges.[7] An empirical criterion chosen in the simulation was to take the final time as three times τ_{max}.

To answer the second question, one must remember that, ideally, the sampling time should be very small if we wanted to simulate a continuous data flow. Practically, there is a trade-off between the approximation of a continuous data flow and the affordable execution time. For the images considered in our simulations, a sampling time of the order of 1 ms yielded good results. In Fig. 7 two simulations with different sampling times are compared. The original image, shown in Fig. 7(a), contains a rectangle of luminance 130 on a background of luminance 124. Object and background are perturbed by additive uniform noise (white noise) to give a SNR of −10 dB. In the simulations shown in Figs. 7(b) and 7(c) the data flow has been sampled with a sampling time of 1 ms and 0.4 ms,

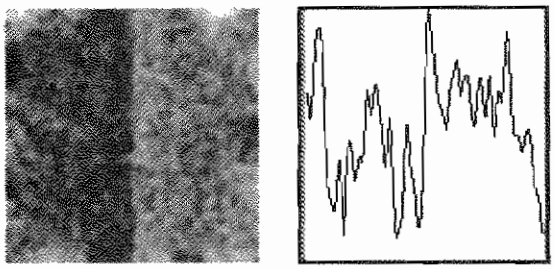

Fig. 5. Result of synchronous processing (at time 20 ms) of the image shown in Fig. 4(a). The right-hand box shows the average of three lines, the same as those defiued in Fig. 4(a). Since all informatiou is processed at the same time, the noise caunot be reduced, and thus the boundary is poorly defined. This resnlt is to be compared with the asynchronous diffusion result presented in Fig. 4(d).

Fig. 6. Labeling. The results of (a) asynchronous and (b) synchronous diffusion at time 20 ms [shown in Figs. 4(d) and 5, respectively] are thresholded. All values smaller than the mean value are interpreted as the background (dark), the other values being interpreted as the foreground (white). Although the contour in (a) is not straight, segmentation is much less noisy in (a) than in (b).

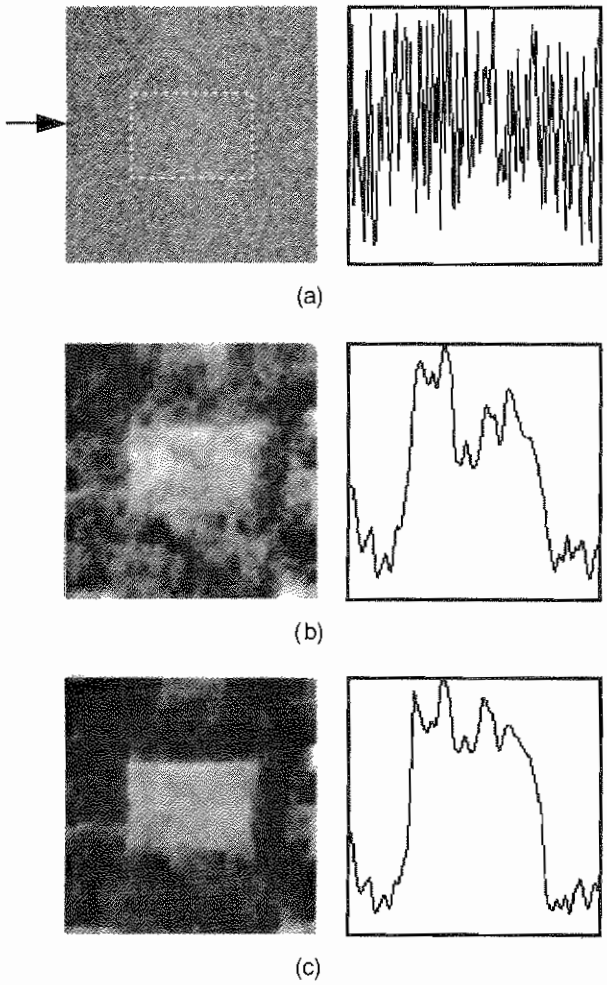

Fig. 7. Effect of sampling of the data flow on the result. Two different sampling times were used. The original image, shown in (a) at the left, has a rectangle of luminance value 130 (delimited by dashed lines) on a background of luminance value 124, both perturbed by additive uniform white noise with a standard deviation of 65.7; the SNR is −10 dB. Results of diffusion at time 15 ms are shown for a sampling time of (b) 1 ms and (c) 0.4 ms. The right-hand column shows line 54, identified by an arrow in the original image. From a comparison of (b) and (c) it is apparent that a data flow sampled with a smaller sampling time gives a better boundary; this result stems from the greater interactiou permitted hetween image points.

respectively (the sampling time of the diffusion equation was the same for these two simulations). The results of the diffusion at time 15 ms demonstrates that a smaller sampling time yields better contours, because the spatial interferences between object and background are correspondingly reduced. An extreme condition is given for a sampling time of value τ_{max} corresponding to the synchronous case, whose result is shown in Fig. 8. This result shows poor boundaries compared with those shown in Fig. 7(c).

The use of a small sampling time implied that the overall processing had to be reiterated many times (the number of iterations being given by the ratio between the final-time and sampling-time values). Also, the execution time for the above simulations was of the order of a few minutes on a Connection Machine, compared with a few hours on a serial machine (Sun sparc station II, 32-MByte memory). The speedup obtained through parallelization for a 128 × 128 pixel image was 30 times for the isotropic filtering, 40 times for the diffusion, 85 times for the competition, and 108 times for the Gabor filtering.

5. OTHER APPLICATIONS

Applications of our model to other classes of images is currently limited by the quality of the edge-estimation stage. In particular, the competition stage does not satisfactorily sharpen edges in complex images. Furthermore, the Gabor filters should be replaced by more-sophisticated cooperative mechanisms. Despite these limitations, we consider a real image of pebbles to illustrate another aspect of our temporal approach. As shown in Fig. 9, the processing of this image was stopped at time 10 ms. At

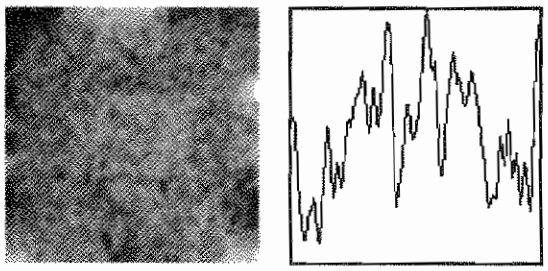

Fig. 8. Result of the synchronous processing of the image shown in Fig. 7(a) (the right-hand box shows line 54). This result represents the limiting case in which the sampling time of the data flow is so coarse that all information is processed at the same time, thus uot allowing for any interaction among image points. The boundary is to be compared with the one obtained with an asynchronous segmentation, as shown in Fig. 7(c).

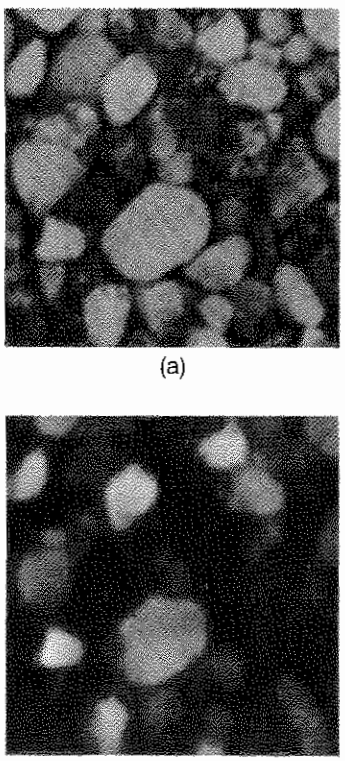

Fig. 9. Asynchronous processing of a real image: (a) a photograph of pebbles (128 × 128 pixels); (b) the result of diffusion at time 10 ms (sampling time 0.3 ms). At this time, only the most-luminous pebbles have been processed.

this time the brightest pebbles appear as isolated objects. This effect, which results from the asynchronous processing, could be used as the basis of a focus-of-atttention mechanism.

Such a principle of attaching a delay to each image primitive could ease the image analysis by pruning out the background as well as objects whose features do not elicit early (or late) responses. This pruning would be beneficial, as it would permit a reduction in the amount of visual information to be processed at each instant.

This principle can be extended to other visual features (such as color) or to other levels of processing. For instance, in previous research, the relationship of curvature to response latency was explored.[22] In addition, an indexing mechanism for object discrimination is currently being developed.[23] On a related issue, an amplitude-to-response-latency relationship was used in a neural model of the winner-take-all function,[24] which was applied to model motion processing in the brain.[25]

6. CONCLUSIONS

Luminance-based segmentation was shown in this paper to improve in the case of additive white noise if the processing of image points is delayed proportionally to their luminance. How can this principle be generalized to other visual cues? To answer this question, we discuss the projection mapping of a two-dimensional spatial image onto a three-dimensional spatiotemporal domain. This projection allows neighboring regions of differing luminances to become separated along the temporal dimension. In the case of the noisy images used in our simulations, it was speculated that such temporal processing is optimum in the sense that the probability of misclassifying an object from a background is minimized. For the general case, intuition dictates that this processing strategy would prevent regions of differing properties from interfering with each other. Because the temporal precedence of visual features coming out of the early stages of a visual system could also be used to attract attention to particular regions perceived first (or second), and thus reduce the amount of visual information to be processed at each time, asynchrony is believed to be a general property that permits optimization of the processing of visual information.

ACKNOWLEDGMENTS

We are grateful to Norberto Grzywacz and Julie Harris for their fruitful comments on this paper and to Zi-Ping Hu for her help in the simulations realized on the Connection Machine. We also gratefully acknowledge the Swiss National Fund for Scientific Research.

Address all correspondence to P.-Y. Burgi.

REFERENCES

1. D. H. Ballard and C. M. Brown, *Computer Vision* (Prentice-Hall, Englewood Cliffs, N.J., 1982).
2. P. J. Besl and R. C. Jain, "Three-dimensional object recognition," Comput. Surv. **17,** 75–145 (1985).
3. P. P. Ohanian and R. C. Duhes, "Performance evaluation for four classes of textural features," Pattern Recognition **25,** 819–833 (1992).
4. Y. I. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," Computer Vision Graphics Image Process. **13,** 222–241 (1980).
5. S. Grossberg, E. Mingolla, and D. Todorovic, "A neural network architecture for preattentive vision," IEEE Trans. Biomed. Eng. **36,** 65–84 (1989).
6. P. Perona and J. Malik, "Scale-space and edge detection nsing anisotropic diffusion," IEEE Trans. Pattern Anal. Mach. Intell. **12,** 629–639 (1990).
7. P. Saint-Marc and J.-S. Chen, "Adaptive smoothing: a general tool for early vision," IEEE Trans. Pattern Anal. Mach. Intell. **13,** 514–529 (1991).
8. P.-Y. Burgi and T. Pun, "Figure–gronnd separation: evidence for asynchronous processing in visnal perception?" Perception **20,** 69 (1991).
9. J. M. Cattell, "The influence of the intensity of the stimulus on the length of the reaction time," Brain **8,** 512–515 (1885).
10. J. T. Enright, "On Pulfrich-illusion eye movements and accommodation vergence during visual pursuit," Vision Res. **25,** 1613–1622 (1985).
11. T. Carney, M. A. Paradiso, and R. D. Freeman, "A physiological correlate of the Pulfrich effect in cortical neurons of the cat," Vision Res. **29,** 155–165 (1989).
12. J. M. Williams and A. Lit, "Luminance-dependent visual latency for the Hess effect, the Pulfrich effect, and simple reaction time," Vision Res. **23,** 171–179 (1983).
13. S. M. Courtney and G. Buchshaum, "Temporal differences between color pathways within the retina as a possible origin of subjective colors," Vision Res. **31,** 1541–1548 (1991).
14. M. W. Greenlee and B. G. Breitmeyer, "A choice reaction time analysis of spatial frequency discrimination," Vision Res. **29,** 1575–1586 (1989).
15. D. Thompson and N. Drasdo, "The effect of stimulus contrast on the latency and amplitude of the pattern electroretinogram," Vision Res. **29,** 309–313 (1989).
16. A. Rosenfeld and A. C. Kak, *Digital Picture Processing* (Academic, New York, 1982).
17. R. J. W. Mansfield and J. G. Daugman, "Retinal mechanisms of visual latency," Vision Res. **18,** 1247–1260 (1978).
18. P.-Y. Burgi, "Understanding the early visual system through modeling and temporal analysis of neuronal structnres," Ph.D. dissertation no. 2536, University of Geneva, Geneva, Switzerland, 1992.
19. J. H. R. Maunsell and J. R. Gibson, "Visual response latencies in striate cortex of the macaque monkey," J. Neurophysiol. **68,** 1332–1344 (1992).
20. J. J. Koenderink and A. J. van Doorn, "Visual detection of spatial contrast: influence of location in the visnal field, target extent and illuminance level," Biol. Cybern. **30,** 157–167 (1978).
21. A. M. Waxman, M. Seibert, R. Cunningham, and J. Wu, "Neural analog diffusion-enhancement layer and spatiotemporal grouping in early vision," in *Advances in Neural Information Processing Systems 1,* D. S. Touretzky, ed. (Morgan & Kaufmann, San Mateo, Calif., 1989), pp. 289–296.
22. P.-Y. Burgi and T. Pun, "Temporal analysis of contrast and geometric selectivity in the early human visual system," in *Channels in the Visual Nervous System: Neurophysiology, Psychophysics and Models,* B. Blum, ed. (Freund, London, 1991), pp. 273–288.
23. J.-M. Bost, R. Milanese, and T. Pun, "Temporal precedence in asynchronous visual indexing," presented at the International Conference on Compnter Analysis of Images and Patterns, Budapest, Hungary, September 13–15, 1993.
24. N. M. Grzywacz and A. L. Yuille, "A model for the estimate of local image velocity by cells in the visual cortex," Proc. R. Soc. London Ser. B **239,** 129–161 (1990).
25. A. L. Yuille and N. M. Grzywacz, "A winner-take-all mechanism based on presynaptic inhibition feedback," Neural Computation **1,** 334–347 (1989).