MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

FINDING SPATIO-TEMPORAL CONTOUR OF MOVING OBJECTS

Kenji NAGAO, Masaki SOHMA, Katsura KAWAKAMI, Michihiro KOBAYAKAWA*
Tokyo Information Systems Res. Lab., Matsushita Industrial Co., Ltd.
4-5-15, Higashishinagawa, Shinagawa-ku, Tokyo 140, Japan
Email: nagao@trl.mei.co.jp
and
Shigeru ANDO
Faculty of Engineering, The University of Tokyo

Abstract

This paper describes a new algorithm to find the contour of a moving object. A distinctive feature of this algorithm exists in its bottom-up architecture throughout low-level and intermediate-level(mid-level) processes. In our algorithm, first a complete set of dimensionless spatio-temporal measures are derived to provide low-level constraints on varying brightness. Then, based on the self and neighboring consistency among these measures, candidate regions of the contour are bounded through spatial relaxation operations. Finally in the mid-level process, these low-level measures in the candidate regions are combined with mid-level constraints on spatial and temporal continuity of moving boundaries. This incorporation is made through a newly proposed dimensionless regularization procedure over the trajectory of the moving boundary. We examine the efficiency of this algorithm through several experiments on real NTSC motion pictures with dynamic background and natural textures.

I. Introduction

A typical approach for boundary detection and segmentation in image sequences firstly estimates the optical flow field and secondly detects the flow discontinuities. Thompson[1] estimated the optical flow field through establishing the correspondence relations between feature points of two consecutive frames. After smoothing the flow field with a Gaussian, he located its discontinuities using a Laplacian operator. Schunck[3] proposed a similar algorithm using a different method of optical flow field determination. Thompson's early work[5] deals with a method in which velocity and intensity information is combined based on a clustering algorithm. But all these approaches share some intrinsic difficulties. First, since changes of brightness are caused by accretion and deletion instead of motion near the occluding boundaries, definition of velocity itself is violated there, thus the information produced on these regions becomes not only meaningless but also harmful to the entire algorithm. Second, local information alone is usually insufficient to determine a flow field especially in regions where spatial changes of intensities are small.

To overcome this problem, Mutch[6] attempted to make use of accretion and deletion for occluding boundary detection. She interpreted a missing correspondence as a token of accretion/deletion, and determined motion boundaries by connecting the positions of them. But a clear difficulty of this work is the requirements of densely locating tokens which is hardly realizable in practical situations. The active contour model called 'Snakes' was also applied to this problem[8]. Although it could locate precise contours in a static image with simple textures[7], its effectiveness to real world pictures with complicated textures and numerous contour candidates has not been confirmed.

In this paper, we try to provide an essential solution to this problem. The salient characteristic of our algorithm is its bottom-up abstracting architecture. In this architecture, both low-level constraints involved in brightness changes and mid-level constraints on geometry and dynamics of the moving boundaries build up, in a step-by-step manner, a logically consistent contour through relaxation and regularization procedures appropriate for each level of constraint. This provides us with flexibility in designing procedures to extract features and to incorporate them in each stage so that they can reflect corresponding constraints most appropriately. In order that features extracted independently in the different levels can be incorporated consistently in the regularization procedures, these features must be so designed that each one expresses a normalized measure and is free from any physical dimensions which lead to dependence on spatio-temporal resolutions, gray-level amplitude and so on. This property provides a drastic improvement in the algorithmic robustness against changes in physical nature of motion pictures.

In the low-level process of our algorithm described in Section III, classification measures of spatio-temporal image functions are extracted locally to reflect physical constraints on brightness. By using those measures, we can restrict possible (interpretable without contradiction) regions of contour. Before proceeding to the mid-level process, the possible areas are further narrowed to some extent based on the neighboring consistency between measures, through a relaxation procedure. In the mid-level process described in Section IV, these measures of positions included in remaining possible areas are incorporated through regularization over the surface which is the trajectory of the object boundary in spatio-temporal 3D space. One criterion for this optimization is the image measures mentioned above, and the other is the smoothness measure of the surface which is reflections of mid-level constraints bounding both shapes and dynamics of moving contour. In Section V, we examine the efficiency of this algorithm through several experiments on real NTSC motion pictures with dynamic background and natural textures.

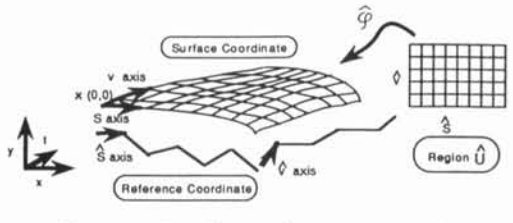

Figure 1: Coordinates for contour patch.

*He was a student of Tokyo Engineering University and participated in the group of Matsushita for this research. Currently he works for Graphtec Corp.

II. Problem Framework

Suppose a boundary of a closed image region in which no motion discontinuity exists. We call a trajectory surface of it in xyt space as *motion contour*. A goal of our algorithm is to find and describe a smooth segment of the motion contour, i.e., a *contour patch*. Then, how the contour patch including its smoothness is described? Here we have three problems: 1) The contour patch considered in this paper is too wide and too bent to be described by a single valued scaler height function[13]. 2) A coordinate system for describing the contour patch (contour coordinate) must be fixed before searching it although its precise position is unknown. 3) Smoothness measures of the contour patch should be invariant with the inaccuracy of the contour coordinate for the consistency and stability of the results. To answer these problems, we consider three kinds of coordinate discussed in the following subsections:

A. Surface coordinate

Let ΔV be a true contour patch and $\mathbf{x} = (x, y, t)$ be a point on ΔV. ΔV is described by grids constructed by an instantaneous contour, i.e., s axis, and some boundary point loci, i.e., v axis. From the assumption of a continuous motion field, there exists a rectangular region ΔU of (s, v) and a bijective map $\varphi : \Delta U \rightarrow \Delta V$ such that $\varphi(s, v) = \mathbf{x}$. Actually, finding ΔU and $\varphi(s, v)$ is a final goal of our algorithm.

B. Reference coordinate

A shape of the contour can be determined by locating sufficient number of points on the contour. A way to specify the points is to construct a reference coordinate near the true contour patch and draw perpendiculars from each mesh points so that they intersect the contour. Reference coordinate $(\hat{s}, \hat{v})$ and a corresponding bijective map $\hat{\varphi}(\hat{s}, \hat{v}) = \mathbf{x}$ are determined in this paper by a roughly estimated grid in a candidate region of contour established by a low-level process. For an always visible boundary, we obtain $\Delta\hat{U}$ as a rectangular region. $\Delta\hat{U}$ is equal to ΔU in our algorithm.

C. Smoothness coordinate

Since the interval of the $(\hat{s}, \hat{v})$ grid is not only variant by motion but also noisy because of uncertainty of low-level processes, it is inappropriate to define the smoothness measure of motion contour on it. Instead, we introduce a new coordinate system (s', v') to evaluate the smoothness. (s', v') is defined only locally at each point on the surface. The direction of (s', v') axes are equal to the tangent of the boundary and the motion vector defined by

$$\mathbf{v} = (v_x, v_y, 1) \tag{1}$$

where (v_v, v_y) is the velocity of the nearest internal point. The units of (s', v') axes are taken identically to that of x, y axis and t axis respectively. Let $\mathbf{x}'(s', v')$ express a point on ΔV in this coodinate.

III. A low-level process: from image field to logical constraints

This section describes the low-level process in which possible regions for a contour are bounded step-by-step based on spatio-temporal classification measures. A particular emphasis is placed on its abstraction capability for the following regularization process.

A. What is the problem in regularization?

Although regularization has been implemented in many early vision systems[11] for determining tokens such as edges[2], optical flow[4, 9], and smooth curves[10], as well as surfaces[13], they still suffer following serious difficulties:

1) Limited choice of an energy function — criteria based on intensity or its gradients used in conventional implementations are insufficient to make use of various knowledges contained in images. This often results in the involvement of unwanted regions such as discontinuities in the regularization which causes not only inprecision[4] but also enormous computation cost [10].

2) Physical dimension of the criteria — energy functions derived from quantities with physical dimensions suffer inconsistency problems in the incorporation process under some changes of observing conditions such as intensity resolution and spatio-temporal resolutions of input pictures[2, 4, 9, 10, 13].

In order to avoid the above mentioned problems, the low-level process proposed by us is designed in such a way to play the following roles:

1) Local but complete classification of an image field — This allows us to classify, including aperture problems, local pixels according to its consistency with boundary interpretation, then bound without any risk a candidate region of the boundary. This provides robustness and reduced computation in regularization.

2) Dimensionless features for classification and regularization — This means that any quantities including criteria are independent with any changes of physical conditions such as varying intensity and resolution under unstable condition of objects. This will also mean that the realized integration is not physical but logical.

In order to realize these functions, we introduce normalized dimensionless measures, which have been originally proposed by Ando[14, 15].

B. A complete set of image field classification measures

Suppose a local covariance matrix of the spatio-temporal intensity gradients is defined as

$$S_{jk} = \int\int\int_{\Gamma} f_j(x, y, t) f_k(x, y, t) dx dy dt \tag{2}$$

where Γ is a neighborhood area around a point (x, y, t), and $f_j, f_k \in \{f_x, f_y, f_t\}$. Since features of intensity changes can be described by the dimensional property of its distribution, we can use covariances of intensity gradients to classify pixels.

[Measures for spatial changes]

Measures to classify spatial variation are introduced, which have been developed in the literature[14].

[Spatial variation measure]

$$P_1 = \frac{(S_{xx} + S_{yy})^2}{(S_{xx} + S_{yy})^2 + \sigma_v^2} \tag{3}$$

[Directional/Non-directional variation measure]

$$P_2 = \frac{(S_{xx} - S_{yy})^2 + 4S_{xy}^2}{(S_{xx} + S_{yy})^2 + \sigma_v^2} \tag{4}$$

Spatial variation measure P_1 features rapid spatial changes of intensity regardless of the dimensional property of the variation. Directional/Non-directional variation measure P_2 reacts if the intensity changes quickly in one direction, i.e., for the 1 D distribution of spatial intensity gradients. σ_v^2 is added for

noise suppression. If noises are caused only by the quantization, we can set σ_v^2 to $0.5^2 \times \Gamma$.

[Measures for temporal changes]

Measures to classify temporal intensity changes are presented. The validity and the derivation of these measures will be presented somewhere later in detail. For reference see Ando[15].

Temporal intensity changes are divided into classes based on the causes of changes, namely, whether they are due purely to motions, or accretions/deletions of points.

[Two-dimensional accretion(deletion)/motion measure]

$$P_3 = \frac{4det[S]}{(S_{xx}S_{yy} - S_{xy}^2 + 2\sigma_s{}^2)(S_{tt} + \sigma_t{}^2)} \tag{5}$$

Two-dimensional accretion(deletion)/motion measure discriminates local regions in image field where spatial changes are two-dimensional, on whether their temporal intensity changes are caused by quick accretion/deletion, i.e., 3D distribution of spatio-temporal intensity gradients or motion of pixels, i.e., 2D distribution.

[One-dimensional accretion(deletion)/motion measure]

$$P_4 = \frac{S_{xx}S_{tt} - S_{xt}^2}{(S_{xx} + \sigma_s)(S_{tt} + \sigma_t)} \tag{6}$$

One-dimensional accretion(deletion)/motion measure features accretion(deletion) or motion in local regions where spatial intensity changes are one-dimensional, i.e., intensities of pixels changes in one particular direction.

C. Clustering and bounding candidate regions

Pixels are divided into 8 classes listed in Table 1, based on the relations between measures derived for each pixel. This is a local classification based simply on dimensional properties of spatial and temporal intensity changes of each pixel. The flow of classification procedure can be represented by a binary search tree, shown in Figure 2, in which $T_i (i = 0, 1, 2, 3, 4)$ are constants for thresholding.

Table 1: Classes based on low-level measures.

S0_FIX	No texture and no brightness change
S0_CHG	No texture and brightness change
S1_FIX	1D texture and no brightness change
S1_ACR	1D texture and inconsistent motion
S1_MOT	1D texture and consistent motion
S2_FIX	2D texture and no brightness change
S2_ACR	2D texture and inconsistent motion
S2_MOT	2D texture and consistent motion

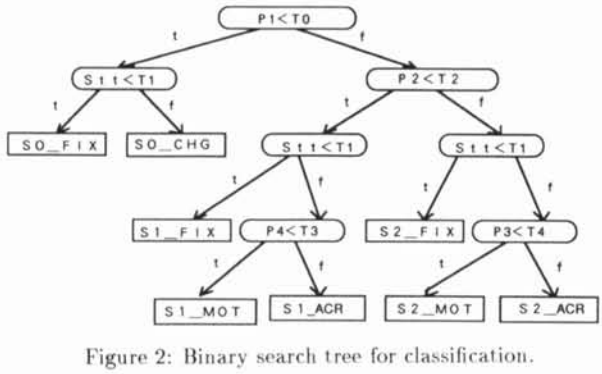

Figure 2: Binary search tree for classification.

Further classification are made on pixels of which first classification are not conclusive, i.e., which are S0_FIX, S0_CHG, S1_FIX, S1_MOT, S1_ACR (other classes are called conclusive classes), based on the consistencies between classes in neighboring regions in image field. The results of this second classification describe whether neighbors around a concerning pixel are involved inside motion regions(S_MOT), boundaries or centers of rotation(S_ACR), stable regions(S_FIX), or such regions impossible to interpret as neighboring intensity changes of pixels do not have any logical consistencies(OTHERS). This second classification is realized through a spatial relaxation algorithm. The local rules for the iterative-improvement employed in this relaxation are based on the relations between the current class given to the concerning pixel and the ratio of conclusive classes involved in its neighbors. See Appendix A. The region which consists of a set of points labeled S_ACR is the candidate region D for the contour.

IV. A mid-level process: incorporating different level constraints to a contour

Under an assumption of natural motion of an object, the smoothness is generally supported about trajectories of boundary points. Although the smoothness of boundary depends on the object shape, possible application of this algorithm in near future validates the assumption of the simplicity of the objective contour. These two assure the smoothness of the contour patch in both s and v directions. In this section, we define a mid-level measure for describing smoothness of the motion contour, and present a method to incorporate low-level and mid-level measures.

A. Extracting a smoothness measure for a contour patch

As a smoothness measure R of the motion contour, we define on (s', v') coordinate a dimensionless function

$$R(\mathbf{x}) = \frac{e_s(\mathbf{x})}{e_s(\mathbf{x}) + \sigma_r^2} \tag{7}$$

of an inhomogeneous quadratic variation

$$e_s(\mathbf{x}) = \{\alpha^4 |\frac{\partial^2 \mathbf{x}'}{\partial s'^2}|^2 + 2\alpha^2 |\frac{\partial^2 \mathbf{x}'}{\partial s' \partial v'}|^2 + |\frac{\partial^2 \mathbf{x}'}{\partial v'^2}|^2 \}_{(s',v')=(0,0)} \tag{8}$$

where σ_r^2 is a term for noise suppression. e_s is a derivation from a newly extended quadratic variation for a surface defined in (x, y, t) coordinate.

The quadratic variation

$$(\frac{\partial^2 z}{\partial x^2})^2 + 2(\frac{\partial^2 z}{\partial x \partial y})^2 + (\frac{\partial^2 z}{\partial y^2})^2 \tag{9}$$

was originally employed by Grimson[13] to measure the smoothness of a scaler depth function $z = f(x, y)$. He also showed its invariance with translation and rotation in xy plane, and discussed analytically and empirically the superiority of this measure to other differential geometrical measures[12].

The basic validity to employ e_s for contour patch ΔV is supported by the equivalency between e_s' defined below and a quadratic variation for a certain depth function expressing the contour patch ΔV in the smoothness coordinate (See Appendix B).

$$e_s' = |\frac{\partial^2 \mathbf{x}'}{\partial s'^2}|^2 + 2|\frac{\partial^2 \mathbf{x}'}{\partial s' \partial v'}|^2 + |\frac{\partial^2 \mathbf{x}'}{\partial v'^2}|^2 \tag{10}$$

Since the definition of e_s' is halfly direct with (x, y, t) coordinate and halfly parametric with (s', v') coordinate, it is computationally easy to evaluate.

Parameter α which is introduced in e_s also plays a significant role for the validity of e_s as well. As described in the introduction to this section, though the smoothness of contour patch ΔV in the direction of v' axis is supported by physical principals, the assumption of smoothness in the direction of s' axis relies simply on the smoothness of the outline curve. This means that unlike in Grimson's case, we should take into account the directional property in designing smoothness measures for the contour patch ΔV. Therefore, in order to realize this aspect as well as that described in Section III, i.e., normalizing measures, we must insert α in e'_s so that the extended quadratic variation can be used for surfaces which have orientation in smoothness and it may have a consistent physical dimension as well. For the derivatives and the validity of $R(\mathbf{x})$ in detail, see Appendix B.

B. Incorporating low-level and mid-level measures through regularization

In this section, possible interpretations of intensity changes obtained in the low-level process are incorporated with mid-level constraints on the surface smoothness to determine a contour. Integration is made over a contour patch in reference coordinate through regularization. The energy function to be minimized is shown in (11), where λ_i are weights.

$$E = -\sum_{i=1}^{4} \lambda_i E_{pi} + E_s \tag{11}$$

The first term is the penalty functional, which is defined as a combination of integrals of low-level classification measures over the contour patch ΔV, each of which being specified in (12).

$$E_{p_i}(\hat{\varphi}) = \iint_{\Delta\hat{U}} P_i(\hat{\varphi}(\hat{s},\hat{v}))d\hat{s}d\hat{v} \quad (i \in \{1,2,3,4\}) \tag{12}$$

where $P_i(x,y,t)$ is a real function of (x,y,t), which estimates the classification measure P_i at point (x,y,t) in spatio-temporal space. P_3, P_4 returns 0 if the spatial intensity change at (x,y,t) does not have corresponding dimensional property, i.e., two-dimensional for P_3, one-dimensional for P_4. The target of energy minimization in regularization can be bounded to the region D which is derived as the possible candidate for contours.

The second term is the stabilizer which constraints the smoothness of the contour patch to be recovered. It is defined as the integration of the smoothness measure R over the range $\Delta\hat{U}$. As already described, the smoothness measure R is defined at each position on the contour patch and its computation is implemented in the local smoothness coordinate. Thus, the stabilizer integrates the smoothness measure estimated in the local system, over a global surface expressed by a reference coordinate system. In the detection of energy minimization, since unlike the smoothness coordinate established on each candidate contour patch, the reference system is an absolute system which is objective for any candidate contour patches, the validity of the establishment of coordinates for describing optimization process is clearly supported.

V. Experimental results

This section shows some experimental results of contour detection on real NTSC video pictures. In order to minimize the energy function (11), we have developed a new hierarchical graph search algorithm. The details of this algorithm will be presented in a succeeding paper.

A. Experiment 1

Results are shown on pictures with dynamic background and complicated textures. In this picture, the background is translating leftward by $1 \sim 2$(pix) per frame interval, which is caused by the camera's motion, and, at the same time, a person is translating leftward by $0.5 \sim 1$(pix) relatively to the background. Figure 3(a) shows the results of local classification based on the classification measures. Parameters T_0, T_1, T_2, T_3, T_4 used for thresholding in the binary search are set respectively to 0.5, 25, 0.125, 0.5, 0.25. Figure 3(b) shows the results of second classification. The scope of integration Γ in (2) is 5(pix) $\times$ 5(pix) $\times$ 2(time-interval). ${\sigma_v}^2$, ${\sigma_s}^2$,${\sigma_t}^2$ were all set to $40 \times 25 \times 2$. Figure 3(c) shows the candidate region D for a contour of the person's area. The approximation of the contour used for $\hat{s}$ coordinate, which is shown in Figure 3(d), was derived through a search of the inner boundary of the candidate region for a contour. In Figure 3(e), the contour segments extracted through our algorithm were overlaid over the source image, without any adjustments between neighboring segments on their terminals. $\Delta\hat{U}$ was taken as $10(\hat{s}) \times 7(\hat{v})$, α was set to 0.7, σ_r^2 was set to 10, and weights in energy function were set respectively $\lambda_1 = 0$, λ_2=2.0, λ_3=5.0, $\lambda_4 = 2.0$. These values were chosen empirically. From Figure 3(e), we can make sure that the contour was exactly detected by our algorithm.

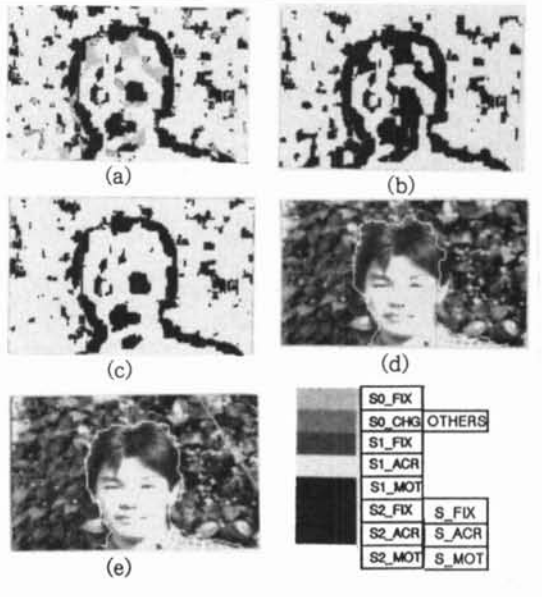

Figure 3: Extracting motion contour from dynamic background.
The size of the picture is 180(pix)$\times$ 120(pix), and intensity resolution is 256 level.

B. Experiment 2

In this experiment, a contour of a moving region in which pixels do not have consistent velocities is detected.

The source picture has 4 times as high resolution as that used in the first experiment. In this picture, the whole field is slowly being magnified by the zooming operation of the camera, and, at the same time, the right hand with a racket is rotating around the right shoulder. The displacement of pixels around the racket area is around $2 \sim 3$(pix) per frame interval, while that around the shoulder is about $0.5 \sim 1$(pix), thus the velocity field does not have a consistent velocity. As the average magnitude of the optical velocity in the hand plus racket area doubles that of person's area extracted in the first experiment, the sampling frequency in time of this picture can

be approximately regarded as half of that of the picture used in the first experiment.

Figure 4(a) shows the results of local classification of pixels. The area Γ as well as the parameter $\sigma_v, \sigma_s, \sigma_t$ for noise suppression and parameters $T_i (i = 0,1,2,3,4)$ for thresholding were all set to the same as those in the first experiment. Results of second classification are shown in Figure 4(b). Figure 4(c) shows the candidate region D for the contour of a moving region which consists of the right arm with the racket. The approximation of the contour used for $\hat{s}$ coordinate is derived in the same way as that in the first experiment, which is shown in Figure 4(d). In Figure 4(e), results of finding the contour segment through regularization are overlaied on the source picture. The parameter α as well as σ_r^2 and the weights $\lambda_i (i = 1,2,3,4)$ were all set to the same values as those in the first experiment. The scope $\Delta\hat{U}$ for which surface detection was made was taken as $10(\hat{s}) \times 7(\hat{v})$.

From Figure 4(e), it is clear that exact detection of the contour has been attained for a moving region in which pixels do not have consistent optical velocities. In addition, although the spatio-temporal resolutions of source pictures have been changed from those in the first experiment, since it was not required to readjust weights in regularization, the effectiveness of normalization of classification measures has been demonstrated.

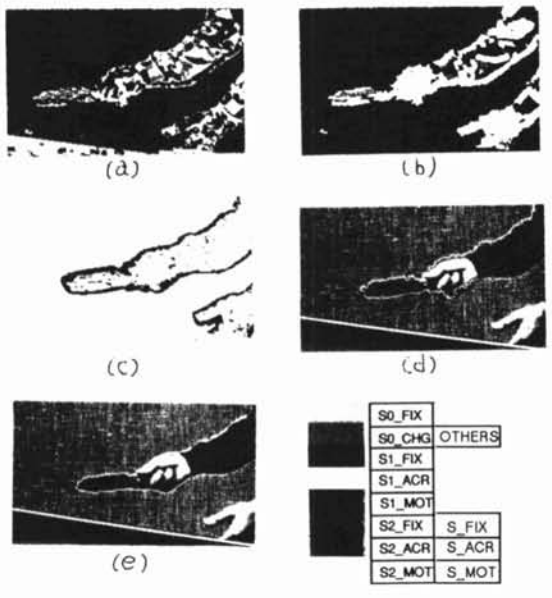

Figure 4: Finding motion contour of an object whose pixels have inconsistent velocities.
The size of the picture is 360(pix) × 240(pix), and intensity resolution is 256 level.

VI. Conclusion

In this paper, a robust algorithm to locate spatio-temporal contours has been proposed, which consists of two abstracting stages of image information, low-level and mid-level abstraction. In the low-level process, possible regions for the contour are bounded based on classification measures composed of covariances of intensity gradients. Succeeding mid-level process determined a contour, integrating these low-level measures with smoothness constraints on motion contour by a new regularization frame work. Some experimental results on real pictures to find contours showed the effectiveness of our algorithm as a practical algorithm. Furthermore, a generalized solution to attack the difficulties with conventional regularization was proposed, and it was examined that our feature extraction is quite appropriate for regularization.

Acknowledgements

We are grateful to Dr. Hiroshi Nishikawa, Dr. Kei Yuasa, Dr. Pradeep Kuma Sinha, and Dr. Hiroyoshi Uno for many stimulating discussions with us.

References

[1] W. B. Thompson, K. M. Mutch and V. A. Berzins, "Dynamic Occlusion Analysis in Optical Flow Fields", IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-7, pp.374-383, 1985.

[2] V. Torre, T. A. Poggio, "On Edge Detection", IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, pp.147-163, 1986.

[3] B. G. Schunck, "Image Flow Segmentation and Estimation by Constraint Line Clustering", IEEE Trans. Pattern Anal. Mach. Intell, vol. PAMI-11, pp.1010-1027, 1989.

[4] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow", Artif. Intell., vol. 17, pp.185-203, 1981.

[5] W. B.Thompson, "Combining Motion and Contrast for Segmentation", IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-2, pp.543-549, 1980.

[6] K. M. Mutch and W. B. Thompson, "Analysis of Accretion and Deletion at Boundaries in Dynamic Scenes", IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-7, pp.133-138, 1985.

[7] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Model", in Proc. ICCV-87, pp.259-268, 1987.

[8] N. Ueda, K. Mase, Y. Suenaga, "A Contour Tracking Method Using Elastic Contour Model and Energy Minimization Approach", Trans. IEICE, vol.J75-D-II, No.1, pp.111-120, 1992 (in Japanese).

[9] E. C. Hildreth, "Computations Underlying the Measurement of Visual Motion", Artif. Intell, vol. 23, pp.309-354, 1984.

[10] U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures", Comm. ACM, vol. 14, pp.335-345, 1971.

[11] T. Poggio, V. Torre and C. Koch, "Computational Vision and Regularization Theory", Nature vol. 317, pp.314-319, 1985.

[12] K. Nomizu and S. Kobayashi, "Foundations of Differential Geometry", vol. 1, Willey-Interscience, 1963.

[13] W. E. L. Grimson, "An Implementation of a Computational Theory of Visual Surface Interpolation", Comput. Gr. Image Process, vol. 22, pp.39-69, 1983.

[14] S. Ando, "A class of Local Image Operators Derived from Curvature of Correlation Function", Trans. Soc. Instru. Cont. Engin., Vol.24, No.10, pp.1016-1022, 1988 (in Japanese)

[15] S. Ando: "Gradient-Based Feature Extraction Operators for the Classification of Dynamical Images", Trans. Soc. Instru. Cont. Engin.,, vol.25, No.4, pp.496-503, 1989 (in Japanese)

Appendix

A. Flow of relaxation

In Figure 5, label(i) is a class label which is currently given to pixel i, and mot(i), emg(i), fix(i) are the numbers of pixels in the neighbors around i, of which current class labels are S_MOT, S_EMG, S_FIX respectively, and N is the size of the neighboring regions.

B. Derivation of smoothness measure e_s

In Figure 6, let $\pi_{\hat{s}\hat{v}}$ be a plane spanned by a smoothness coordinate at $\mathbf{x}(\hat{s},\hat{v})$. From the definition, an origin of this plane is $P_0 = \mathbf{x}'(0,0)$, and $\mathbf{x}'(\delta s', \delta v')$ is a position of P_1 of which s', v' coordinate is $(\delta s', \delta v')$. Let P_2 denote a point on $\pi_{\hat{s}\hat{v}}$ such that P_1P_2 is perpendicular to $\pi_{\hat{s}\hat{v}}$. Then, a length of P_1P_2 defines a scaler function ξ of $\delta s', \delta v'$, and its quadratic variation is defined as

$$e_s'' = (\frac{\partial^2 \xi}{\partial s'^2})^2 + 2(\frac{\partial^2 \xi}{\partial s' \partial v'})^2 + (\frac{\partial^2 \xi}{\partial v'^2})^2 \qquad (13)$$

Here, we find a following lemma holds:
[lemma 1]

$$|\frac{\partial^2 \xi}{\partial k_i k_j}| = |\frac{\partial^2 \mathbf{x}'}{\partial k_i k_j}|, \quad i,j \in \{1,2\}, k_i, k_j \in \{s', v'\} \qquad (14)$$

where $|\cdot|$ denotes the norm of a vector. This lemma assures that e_s' defined in (10) is equivalent to e_s''. Introducing a scale transformation $s'' = \alpha s'$ to maintain a consistent dimension as well as the compatible smoothness between s' and v' axis, we obtain a final form of the measure (13) in (8).

Applying a similar procedure for (3), a normalized dimensionless smoothness measure for contour patch ΔV is then derived from (8).

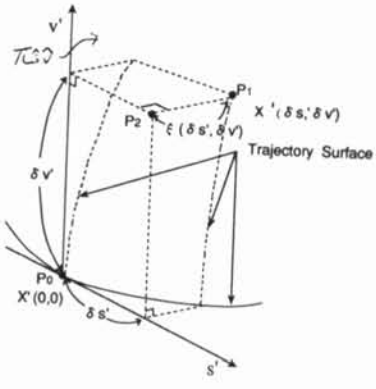

Figure 6: Local function ξ expressing the contour patch ΔV in smoothness coordinate s', v'.

```
procedure relaxation ()
 {
  for each pixel i {
        case : label(i)=S2_FIX
                label(i):=S_FIX;
        case:  label(i)=S2_ACR
                label(i):=S_ACR;
        case:  label(i)=S2_MOT
                label(i):=S_MOT;
   }
   do {
     c=0;
     for each pixel i {
       case: label(i)=S0_FIX or label(i)=S1_FIX
               case: fix(i)>0 and fix(i)>=mot(i) and fix(i)>=emg(i)
                       label(i):=S_FIX;  c++;  break;
               case: mot((i)>N/3 and mot(i)>=fix(i) and mot(i)>=emg(i)
                       label(i):=S_MOT; c++;  break;
               case: acr(i)>N/3 and acr(i)>=fix(i) and acr(i)>=mot(i)
                       label(i):=S_ACR; c++;  break;
       case: label(i)=S1_MOT
               case: mot(i)>0 and mot(i)>=acr(i)
                       label(i):=S_MOT;  c++; break;
               case: acr(i)>N/3 and acr(i)>mot(i)
                       label(i):=S_ACR; c++; break;
        case: label(i)=S1_ACR
               case: acr(i)>0 or (fix(i)>0 and mot(i)>0)
                       label(i):=S_ACR; c++; break;
       default:      break;
     }
   }   while(c>0);
     for each pixel i {
        if (label(i)=S1_FIX or label(i)=S1_MOT or label(i)=S1_ACR) {
          label(i)=OTHERS;
        }
     }
}
```

Figure 5: Flow of relaxation described in C-like code.