Object's tracking by advection of a distance map

Yann Lepoittevin, Isabelle Herlin, Dominique Béréziat

OBJECT'S TRACKING BY ADVECTION OF A DISTANCE MAP

Yann Lepoittevin[*][*]   *Isabelle Herlin*[*]   *Dominique Béréziat*[†]

[*]Inria
CEREA, joint laboratory École des Ponts ParisTech - EDF R&D, Université Paris-Est
B.P. 105, 78153 Le Chesnay, France
[†]LIP6
Université Pierre et Marie Curie
4 place Jussieu
75005 Paris France

ABSTRACT

This paper has two coupled objectives: estimating motion and tracking a given object on an image sequence. It relies on a data assimilation approach, that solves evolution equations of motion, those of image brightness, and those of the distance map modeling the object's boundary. The two last express the optical flow constraint, which assumes that image brightness and distance map are advected by velocity. The method assimilates contour points by an innovative approach combining two criteria. First, the boundary of the object should match contour points at acquisition dates; second, the control of the distance between each pixel and the object's boundary allows to better motion estimation on the whole domain. The method is tested on synthetic data and satellite acquisitions.

Index Terms— Advection, Data Assimilation, Distance Map, Motion Estimation, Object's Tracking.

1. INTRODUCTION

The paper concerns the problem of estimating motion on an image sequence while simultaneously detecting and tracking a given object on the sequence. Yilmaz et al. [1] provide an extensive description about research on object's tracking. As it has been analyzed in Lepoittevin et al. [2], processing noisy images, such as satellite acquisitions, benefits of involving in the approach assumptions on the dynamics. Methods exist that segment and track a structure, given motion field on the studied temporal window and initial segmentation [3, 4, 5], or that track a structure and estimate its displacement if the structure has been previously accurately segmented on the first image [6]. In this paper, we describe an approach that relies on data assimilation and simultaneously solves the issues of motion estimation on the whole image sequence and of tracking an object. The motion field is implicitly related with the tracking trough the temporal evolution model: the

assimilation technique exploits the information contained in the tracking process to better the motion estimation. The object is characterized by its boundary $C(t)$, that is initialized as a closed curve C_0 roughly segmenting the object on the first image. It is modeled by the signed distance map $\phi(t)$ to $C(t)$. The evolution model involved in the data assimilation system assumes the Lagrangian constancy of velocity $\mathbf{w}(t)$ and the transport of the image brightness $I(t)$ and distance map $\phi(t)$ by the motion field $\mathbf{w}(t)$. The outputs of the model are compared to image observations, at each acquisition date, and optimization on their discrepancy is performed to obtain the result on motion and tracking. As the control of the distance map is used to better the motion estimation, advection has to preserve distance properties. This is a key point as most numerical schemes immediately lose the distance map property when advecting a function that is initialized by a distance map at $t = 0$. A reinitialization module is then defined to impose the distance map property during the temporal integration. One major constraint for conceiving that module is to derive an efficient implementation of its adjoint: this adjoint impacts motion estimation during the backward integration of 4D-Var data assimilation [7]. At each date t, $\phi(t)$ is obtained as the result of advecting the distance map $\phi(t-1)$ by the motion field $\mathbf{w}(t-1)$. The gradient property of $\phi(t)$, for being a distance map, should be valid: $||\nabla\phi(\mathbf{x}, t)|| = 1$ for all $\mathbf{x}$, with ∇ denoting the spatial gradient. Let consider $C(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$ the boundary of the tracked object. An upper bound s of the displacement of $C(t)$ from t to $t + 1$ may be defined from heuristics, and $C(t + 1)$ should belong to the region $\Omega_s(t)$, of size s, surrounding $C(t)$: $\Omega_s(t) = \{\mathbf{x} \mid |\phi(\mathbf{x}, t)| < s\}$. The gradient property of $\phi(t)$ has then to be verified only within $\Omega_s(t)$ and not on the whole image domain. Moreover, as the object is initialized by a closed curve C_0, $C(t)$ has to remain a closed curve. Two major approaches are at hand for defining the reinitialization. The first approach consists in defining, from the velocity field $\mathbf{w}(t-1)$, a modified velocity field,

---

denoted $\mathbf{w}_m(t-1)$, that advects $\phi(t-1)$ so that $\phi(t)$ satisfies the gradient property in $\Omega_s(t)$. Such a modified velocity field $\mathbf{w}_m(t-1)$ may be obtained with the Fast Marching method defined by Sethian [8]. Using the spatial discretization schemes defined in [8], $C(t)$ satisfies the property of being a closed curve. This Fast Marching method has been evaluated as a candidate to be involved in our data assimilation system. However its adjoint needs to store a table with the order in which pixels are processed at each time step. The data assimilation requires a large number of calls to that table. The resulting memory requirement and computation cost would have been too expensive and we gave up the idea of computing this modified velocity field. The second approach relies on the method proposed in [9]: $\phi(t-1)$ is first advected by $\mathbf{w}(t-1)$, then, as the result is no more a distance map, it is modified to retrieve the gradient property $(\|\nabla\phi(\mathbf{x},t)\| = 1)$ in the region $\Omega_s(t)$.

Section 2 describes the main mathematical components of the approach, while Section 3 focuses on the reinitialization process. Section 4 discusses results obtained on synthetic data and meteorological satellite acquisitions. Section 5 concludes with some remarks and perspectives on the research work.

2. MATHEMATICAL SETTING

Ω denotes the bounded image domain, $[0,T]$ the studied temporal interval, and $A = \Omega \times [0,T]$. $\mathbf{w}$ denotes the motion function defined on A. As a consequence, $\mathbf{w}(t)$ is a function on Ω and $\mathbf{w}(\mathbf{x},t)$ is the value of point $\mathbf{x}$ at time t. The same remark is valid for all mathematical quantities used in the paper.

Observations, that are available to estimate motion and track the studied object, are images and their contour points, obtained by thresholding the maxima of the gradient norm in the direction of the gradient vector [10].

2.1. Evolution model

The assumption on dynamics is the Lagrangian constancy of velocity $\mathbf{w}$, rewritten as:

$$\frac{d\mathbf{w}}{dt} = \frac{\partial\mathbf{w}}{\partial t} + (\mathbf{w}.\nabla)\mathbf{w} = 0 \qquad (1)$$

A pseudo-image I_s is defined, that satisfies the optical flow constraint:

$$\frac{\partial I_s}{\partial t} + \nabla I_s.\mathbf{w} = 0 \qquad (2)$$

The pseudo-image function is compared to image acquisitions during the optimization process: they have to be almost identical. The function ϕ is assumed to satisfy the same heuristics, as the object moves accordingly to images:

$$\frac{\partial\phi}{\partial t} + \nabla\phi.\mathbf{w} = 0 \qquad (3)$$

The state vector, defined as $\mathbf{X} = \begin{pmatrix} \mathbf{w} & I_s & \phi \end{pmatrix}^T$, satisfies the evolution system (1, 2, 3), summarized by:

$$\frac{\partial\mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}(t)) = 0 \qquad (4)$$

2.2. 4D-Var data assimilation

In order to estimate $\mathbf{X}$, and obtain motion estimation and object's tracking, 4D-Var considers the following three equations:

$$\frac{\partial\mathbf{X}}{\partial t}(\mathbf{x},t) + \mathbb{M}(\mathbf{X})(\mathbf{x},t) = 0 \qquad (5)$$
$$\mathbf{X}(\mathbf{x},0) = \mathbf{X}_B(\mathbf{x}) + \mathcal{E}_B(\mathbf{x}) \qquad (6)$$
$$\mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x},t) = \mathcal{E}_R(\mathbf{x},t) \qquad (7)$$

Eq. (5) is the evolution equation: $\mathbf{X}(\mathbf{x},t)$ is determined, for any t, from $\mathbf{X}(\mathbf{x},0)$. Eq. (6) corresponds to the knowledge, the background value $\mathbf{X}_B(\mathbf{x})$, that is available on the state vector at initial date 0. An error term, $\mathcal{E}_B(\mathbf{x})$, is added in order to express uncertainty. No information being available on velocity, its background value is taken null; the background on the pseudo-image I_s is the first image of the sequence; and the background of ϕ, denoted ϕ_B, roughly defines the object to be tracked. If $\mathbb{P}$ denotes the projection of $\mathbf{X}$ on I_s and ϕ, Eq. (6) is rewritten as:

$$\mathbb{P}(\mathbf{X}(0)) = \mathbb{P}(\mathbf{X}_B) + \mathcal{E}_B \qquad (8)$$

Eq. (7) is the observation equation, that links the observations to the state vector $\mathbf{X}$. The observation vector $\mathbf{Y}$ includes images that are processed to estimate motion. Contours are computed on these images. Distance maps to these points, denoted $D_c(\mathbf{x},t)$, are also included in the observation vector $\mathbf{Y}$. The observation operator $\mathbb{H}$ is split into two parts: $\mathbb{H} = \begin{pmatrix} \mathbb{H}_I & \mathbb{H}_\phi \end{pmatrix}^T$. $\mathbb{H}_I$ compares pseudo-images I_s to image observations I:

$$\mathbb{H}_I(\mathbf{X},\mathbf{Y}) = I_s - I = \mathcal{E}_I \qquad (9)$$

Their discrepancy is described by the error $\mathcal{E}_I$. $\mathbb{H}_\phi$ compares ϕ to the distance map $D_c(\mathbf{x},t)$. The absolute value of ϕ should be almost equal to D_c. As, $C(t)$ is supposed to move from maximum s pixels at each time step, $C(t)$ stays in the region $\Omega_s(t)$ and the comparison is only needed there. A function ζ_s is then defined on Ω such that $\zeta_s(\mathbf{x}) = 1$ near $C(t)$ and decreases continuously to $\zeta_s(\mathbf{x}) < 10^{-5}$ outside $\Omega_s(t)$. The observation operator is written as:

$$\mathbb{H}_\phi(\mathbf{X},\mathbf{Y}) = \zeta_s(\mathbf{x})\left(|\phi| - D_c\right) = \mathcal{E}_\phi \qquad (10)$$

Errors $\mathcal{E}_B$, $\mathcal{E}_I$, $\mathcal{E}_\phi$ are supposed Gaussian, zero-mean and not correlated. The covariance matrix of $\mathcal{E}_B$ is denoted B, and R_I and R_ϕ are the variances of $\mathcal{E}_I$ and $\mathcal{E}_\phi$. Solving System

(5, 8, 9, 10) is then written as the minimization of a cost function, with control $\mathbf{X}(0)$:

$$J(\mathbf{X}(0)) = \int_A \frac{\mathcal{E}_I(\mathbf{x},t)^2}{R_I(\mathbf{x},t)} d\mathbf{x} + \int_A \frac{\mathcal{E}_\phi(\mathbf{x},t)^2}{R_\phi(\mathbf{x},t)} d\mathbf{x}$$
$$+ \int_\Omega \mathcal{E}_B(\mathbf{x})^T B(\mathbf{x})^{-1} \mathcal{E}_B(\mathbf{x}) d\mathbf{x} \quad (11)$$

An adjoint variable λ is defined by:

$$\lambda(T) = 0 \quad (12)$$
$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda = \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1} \mathbb{H}(\mathbf{X}, \mathbf{Y}) \quad (13)$$

The adjoint operators $\left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ and $\left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$ are defined according to: $\langle Z\eta, \lambda \rangle = \langle \eta, Z^*\lambda \rangle$. $\left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$ is mathematically computed and $\left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ is automatically generated from the discrete operator $\mathbb{M}$ by the differentiation software Tapenade [11]. Gradient of J is obtained as:

$$\frac{\partial J}{\partial \mathbf{X}(0)} = \mathbb{P}^T B^{-1} [\mathbb{P}(\mathbf{X}(0)) - \mathbb{P}(\mathbf{X}_B)] + \lambda(0) \quad (14)$$

Minimization is achieved with the L-BFGS algorithm [12].

3. REINITIALIZATION

At date t, let denote ϕ_t the value of $\phi(t)$ that is obtained when advecting $\phi(t-1)$. As the result is no more a distance map, a reinitialization is implemented in order to retrieve that property. The reinitialization process introduced in Sussman et al. [9] consists to find the steady state ψ_l of the following partial differential equation:

$$\frac{\partial \psi}{\partial \tau} = \text{sign}(\phi_t)(1 - ||\nabla \psi||) \quad (15)$$

with initial condition : $\psi(0) = \phi_t$. It can be seen that the steady state ψ_l is a distance map that verifies $||\nabla \psi_l(\mathbf{x})|| = 1$, for all $\mathbf{x}$. If the process also forces ψ_l to have the same zero values than ϕ_t, then ψ_l is the distance map characterizing $C(t)$, and $\phi(t)$ gets the value ψ_l before further integration from t to $t+1$ (this issue of zero value is discussed in the next paragraph). However, as previously stated, the gradient property of ϕ has only to be obtained in the region $\Omega_s(t)$ surrounding $C(t)$. Therefore, integration of Eq. (15) is stopped when the property $||\nabla \psi_l(\mathbf{x})|| = 1$ is valid for $|\psi_l(\mathbf{x})| < s$. This means, according to Sussman et al. [9], that one only needs $\frac{s}{\Delta \tau}$ integration steps of Eq. (15), $\Delta \tau$ being the time step used for discretizing Eq. (15). As the number of integration steps is known in advance, the adjoint used by 4D-Var may be efficiently implemented with a reasonable computing cost and memory requirement.

In the continuous case, Sussman et al. [9] demonstrated that ψ_l has the same zero value curve $C(t)$ than ϕ_t. However, this is no more true during numerical computation due to discretization. Let denote C_ε a small narrow band of size 2ε surrounding $C(t)$. Let P_{ij} denote a pixel with indices i and j. The solution is to force the value ψ_{ij} of ψ to stay close to that of $\phi_{t,ij}$, for each P_{ij} intersecting C_ε, during integration of (15). In the same way, the value of $||\nabla \psi_{ij}||$ in C_ε should stay close to that of $||\nabla \phi_{t,ij}||$. For that purpose, a constraint $\mu f(\phi_t)$ is added to the right term of (15) so that $\text{sign}(\phi_t)(1 - ||\nabla \psi||) + \mu f(\phi_t)$ is almost equal to zero in C_ε. Equation (15) is then rewritten as:

$$\frac{\partial \psi}{\partial \tau} = \text{sign}(\phi_t)(1 - ||\nabla \psi||) + \mu f(\phi_t) \quad (16)$$

Before defining μ and f, it should be noted that the region C_ε is defined by an approximation of the indicator function, that is the derivative of H_ε with:

$$H_\varepsilon(\psi) = \begin{cases} 0 & \text{if } \psi < -\varepsilon \\ \frac{1}{2}\left(1 + \frac{\psi}{\varepsilon} - \frac{1}{\pi}\sin(\pi\frac{\psi}{\varepsilon})\right) & \text{if } |\psi| < \varepsilon \\ 1 & \text{if } \psi > \varepsilon \end{cases} \quad (17)$$

Keeping constant the values of $||\nabla \psi_{ij}||$ in C_ε is then equivalent to keep constant the values of $H'_\varepsilon(\psi)||\nabla \psi_{ij}||$ with the discrete value being defined as the integral of the continuous function within pixel P_{ij}:

$$H'_\varepsilon(\psi)||\nabla \psi_{ij}|| = \frac{\int_{P_{ij}} H'_\varepsilon(\psi)||\nabla \psi(\mathbf{x})||d\mathbf{x}}{\int_{P_{ij}} H'_\varepsilon(\psi)d\mathbf{x}} \quad (18)$$

$f(\phi_t)$ gets value $||\nabla \phi_t||$. The value of μ is defined so that the right term of (16) is almost null in C_ε :

$$\mu = \frac{-\int_{P_{ij}} H'_\varepsilon(\phi_t)\big(\text{sign}(\phi_t)(1 - ||\nabla \psi||)\big)d\mathbf{x}}{\int_{P_{ij}} H'_\varepsilon(\phi_t)f(\phi_t)d\mathbf{x}} \quad (19)$$

This reinitialization module is added to the evolution equation of ϕ in Eq. (3). After discretization, one integration step of the evolution model (5) is:

1. Advection of $\mathbf{w}(t)$, $I_s(t)$, $\phi(t)$ by $\mathbf{w}(t)$, which gives $\mathbf{w}(t+1)$, $I_s(t+1)$, ϕ_{t+1}

2. Given ϕ_{t+1} as initial value, $\frac{s}{\Delta \tau}$ steps of reinitialization are applied for the evolution of ψ

3. $\phi(t+1) = \psi_l$

4. RESULTS

4.1. Twin experiment

A sequence of eighteen image observations, $I_i = I(t_i)$ for $i = 1$ to 18, is generated by integrating model $\mathbb{M}$, in Eq. (4), from initial conditions displayed in Figure 1. Contours are computed on images I_i and the distance maps $D_c(\mathbf{x}, t_i)$ are then derived. This simulation also provides ground-truth of
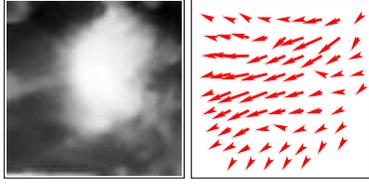
Fig. 1. Left : initial image. Right : initial motion field.

the velocity field. The assimilation method described in 2.2 is applied in order to estimate motion and track the cloud from images I_i. Statistics are computed on the discrepancy between estimated motion $\mathbf{w}_e$ and ground-truth $\mathbf{w}_r$ and given in the first column of Table 1. The second column provides

	with Reinit	without Reinit
Relative norm (in %):	12	18.8
Angular (in degrees)	9.6	12.5

Table 1. Statistics on the errors between estimated motion ($\mathbf{w}_e$) and ground-truth ($\mathbf{w}_r$).

statistics obtained if motion is estimated without including the reinitialization module in the model.

The result is compared with the optical flow method from Sun et al. [13]. The solution cannot be compared immediately, as our method provides a motion field at each step during advection of I_s and ϕ whereas optical flow methods provides a unique motion field between two observations. For illustration purpose, we chose 8 points on the boundary of the object, that are followed during advection. Figure 2 displays the position of these 8 points at final date. The error mean,
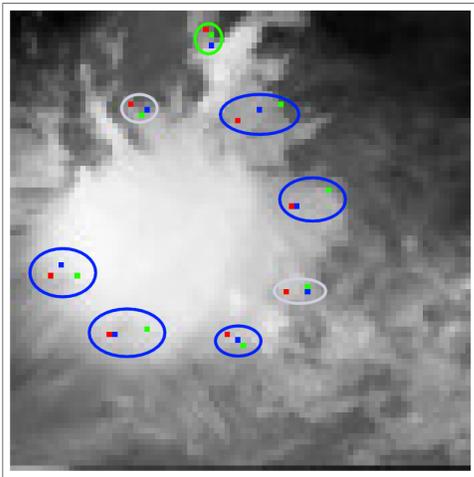

Fig. 2. Red point: ground truth. Blue point: our method. Green point: optical flow from Sun et al. [13] . Blue ellipse: our method is the best. Green ellipse: Sun's result is the best. Grey ellipse : results are equivalent.

computed on more than 200 points, is 2.2 pixels with our approach whereas it is above 4.6 pixels for each tested optical flow method ([13], [14], [15]).

4.2. Meteosat images

The assimilation method is applied on a Meteosat sequence of 18 images. Results of tracking a tropical cloud are given on Figure 3 for 3 images of the sequence. Figure 4 illustrates the
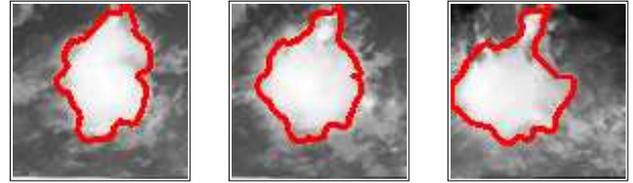

Fig. 3. Tracking a tropical cloud. From left to right, frame: 3, 9, 18.

reinitialization module on a sub-window of the image domain. The result $||\nabla\phi_1||$ is displayed on the left, with ϕ_1 obtained
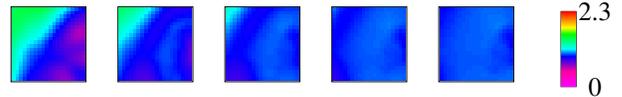

Fig. 4. Iterations of the reinitialization module. Recovering of the gradient property for a subwindow.

when advecting $\phi(0)$. ϕ_1 is not a distance map and its gradient is not uniformly equal to 1 (corresponding to the blue color). The four next images from left to right display $||\nabla\psi||$ during iterations of the reinitialization module. As it can be seen, the gradient property is recovered on the last image, at the end of the reinitialization process.

5. CONCLUSION

This paper describes a method for computing motion on an image sequence and simultaneously tracking a given object on it. The approach is based on a 4D-Var data assimilation technique and the object is modeled by the distance map to its boundary. The dynamic model used for data assimilation assumes Lagrangian constancy of velocity and transport of images and distance maps. The paper explains why the control of the distance, between each pixel and the tracked object's boundary improves motion estimation. Perspectives to this research concern multi-objects tracking and spatio-temporal segmentation.

6. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4 2006, pp. 13, 2006.

[2] Y. Lepoittevin, D. Béréziat, and I. Herlin, "Continuous tracking of structures from an image sequence," in *International Conference on Computer Vision Theory and Applications*, 2013.

[3] N. Peterfreund, "Robust tracking of position and velocity with kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, June 1999.

[4] Y. Rathi, N. Vaswani, A. Tannenbaum, and A.J. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, August 2007.

[5] C. Avenel, É. Mémin, and P. Pérez, "Tracking closed curves with non-linear stochastic filters.," in *Conference on Space-Scale and Variational Methods*, 2009.

[6] M. Bertalmío, G. Sapiro, and G. Randall, "Morphing active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 733–737, July 2000.

[7] F. Le-Dimet and O. Talagrand, *Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects.*, pp. 97–110, Tellus, 1986.

[8] J.A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.

[9] M. Sussman and E. Fatemi, "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *Society for Industrial and Applied Mathematics*, 1999.

[10] John Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[11] L. Hascoët and V. Pascual, "Tapenade 2.1 user's guide," Technical Report 0300, INRIA, 2004.

[12] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal, "L-bfgs-B: Algorithm 778: L-bfgs-B, FORTRAN routines for large scale bound constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.

[13] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of European Conference on Computer Vision*, 2010, pp. 2432–2439.

[14] T. Corpetti, E. Mmin, and P. Prez, "Dense estimation of fluid flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 365–380, Mar. 2002.

[15] B.K.P. Horn and B.G. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.