Multi-Image Edge Detection for SST Images

JEAN-FRANÇOIS CAYULA

Graduate School of Oceanography, University of Rhode Island, Narragansett, Rhode Island
Planning Systems Incorporated, Stennis Space Center, Mississippi

PETER CORNILLON

Graduate School of Oceanography, University of Rhode Island, Narragansett, Rhode Island

(Manuscript received 10 March 1994, in final form 1 December 1994)

ABSTRACT

This paper presents an approach based on the analysis of an image sequence to detect temperature fronts in a sea surface temperature image. The multi-image edge detection algorithm starts by applying a single-image edge detection algorithm to the sequence of images under study. Next, fronts or portions of fronts, which were detected in neighboring images by the single-image algorithm and which match features in the current image, are identified as persistent. The coordinates of these persistent fronts are then passed to the single-image edge detection algorithm so that additional fronts can be detected. The performance of the multi-image edge detection algorithm, of various single-image algorithms, and of a human expert are evaluated on a set of 98 images. For that purpose, the location of the fronts obtained by applying various methods to the SST images is compared to the in situ measures of the Gulf Stream position. With respect to both quality and the number of detected edges, the multi-image edge detection algorithm is the only automated method that achieves results comparable to those obtained by a human expert.

1. Introduction

Satellite-derived sea surface temperature (SST) fields are a major source of information for the study of oceanographic processes. Because these data are available in digital form, there is considerable interest in designing and using algorithms to automatically process SST images. The appearance of more powerful yet less expensive computers has made automating the processing of SST images even more attractive. Of particular concern to oceanographers is the accurate location of edges associated with oceanographic features (Pratt et al. 1991; Podestá et al. 1993). Edge detection is a good candidate for automation for several reasons. Besides the tediousness of the operation, the volume of available data may rapidly become too large for all the processing and analysis to be done manually (Cornillon et al. 1987). Another problem is that subjective results lack consistency: they vary from day to day and person to person. This may make it difficult for oceanographers to compare their findings. By contrast, a single algorithm will always process images the same way and obtain the same results and, thus, can be used as a standard of comparison.

Corresponding author address: Dr. Jean-François Cayula, Graduate School of Oceanography, University of Rhode Island, Narragansett Bay Campus, Narragansett, RI 02882-1197.

However, no algorithm (Cayula and Cornillon 1992; Holyer and Peckinpaugh 1989; Gerson and Gaborski 1977) has yet managed results as good as those obtained subjectively by human operators (Cayula and Cornillon 1992; Cayula et al. 1991; Cornillon and Watts 1987). Subjective front detection, despite a lack of consistency, still outperforms existing algorithms. This manuscript presents an algorithm, which takes advantage of the information contained in a sequence of images to improve the performance of a recent single-image edge detection algorithm (Cayula and Cornillon 1992).

A brief exposé on the particular difficulties associated with SST fields and existing multi-image algorithms, followed by an overview of the algorithm developed in this work, conclude these introductory remarks. Subsequent sections deal with the details of the algorithm.

a. Multi-image processing

Although multi-image processing, such as motion estimation, is an active domain of research, most of that research is not directly applicable due to the particular conditions imposed by SST fields. In general, SST fields are more complex than the images studied in past research on image sequence analysis. Three types of difficulties are encountered when processing sequences of SST images:

1) Because oceanographic objects are not rigid, matching can never be as exact as when dealing with rigid objects. The lack of rigidity also makes it impossible to extrapolate entire oceanographic fronts. This has two important consequences. First, one cannot rely on just a few control points (Snyder 1989) to simplify matching. Second, partially hidden objects, which are a common occurrence in SST fields due to clouds or missing data, become difficult to match from image to image.

2) The singularity assumption (Kahn 1987) does not always hold true. For example, the sharp temperature gradient defining the northern edge of the Gulf Stream is often defined by several temperature steps or fronts. These fronts move following the Gulf Stream meanders at about 10 km per day. With a temporal resolution of 12 h, a point on the northern edge of the Gulf Stream can usually be found within a 10 km × 10 km window centered at the position of the corresponding point in the previous image. However, it is often not possible to determine which of the several fronts in the previous image corresponds to the point of interest. Additional processing is then required to remove the ambiguity and obtain the correct correspondence.

3) The temporal resolution is not sufficient to track certain events. Warm surges (Gilman and Rothstein 1992), which move at speeds of about 100 km per day, cannot be tracked reliably because they lack persistency at the current timescale of one image every 12 h. This situation is made worse because SST data in the Gulf Stream region is corrupted by clouds at least 50% of the time, thus, further degrading temporal resolution.

b. *Overview of the algorithm*

The multi-image edge detection algorithm is actually a series of algorithms. These algorithms include single-image edge detection, persistent contour identification, single-image cloud detection, multi-image cloud detection, and the automatic selection of the Gulf Stream northern edge for the SST fields off Cape Hatteras. This paper only describes in detail the actual edge detection process.

The algorithm starts by applying the single-image edge detection algorithm (Cayula and Cornillon 1992) to a sequence of images (section 2a). Then, the map of persistent contours for the current image is constructed (section 2b). This map includes all fronts or portions of fronts that were detected by the single-image algorithm in neighboring images and correspond to features in the current image. After thinning the contours in the maps of persistent contours (section 2c), the single-image edge detection is applied again to the sequence of images with both the images and their associated maps of persistent contours as inputs (section 2d). The use of persistent contours allows for the detection of weak contours or contours in cloud-corrupted regions. To improve results further, the process of finding persistent contours is repeated with the contours detected in the previous step as input. The revised maps of persistent contours along with the cloud masks obtained from the multi-image cloud detection algorithm (Cayula and Cornillon 1994) are then used when the single-image algorithm is applied a third time to the sequence of images. Multi-image-derived cloud masks are only used in the final step because using these masks earlier would prevent the identification of persistent contours that occur in cloud-corrupted regions. The cloud masks are also unnecessary to the process of identifying persistent fronts because the nonpersistent nature of most clouds means that clouds are unlikely to generate features similar to persistent fronts. The flowchart of the algorithm is presented in Fig. 1.

c. *The data*

The satellite data were obtained from the Advanced Very High Resolution Radiometer (AVHRR) of the *TIROS-N* series satellites (Schwalb 1978). Following navigation, the data were remapped to a common projection. The SST field was calculated using the two-channel algorithm of McClain et al. (1983), which operates on channels 4 and 5 (10.5–11.3 μm; 11.5–12.5 μm). The representation of temperatures in the images is 8 digital counts for 1°C. The images used during development had a spatial resolution of 2 km per pixel and covered an area from 33° to 44°N and 76° to 63°W. The images in the validation set have a spatial resolution of 1 km per pixel and covered an area from 34° to 39°N and 76° to 69°W. Additional information about the data can be found in Cornillon et al. (1987).

2. The algorithm

a. *Single-image edge detection*

The following is a brief overview of the single-image edge detection algorithm described in detail in Cayula and Cornillon (1992) (Fig. 2). This algorithm operates at three levels:

1) The picture level where a statistic, such as the probability that a presegmented region is actually a cloud, is computed for the entire image or for a region in that image.

2) The window level in which the statistic refers to an entire window, for example, the probability that the window contains an SST front.

3) The local level where a pixel is assigned a statistic based on a neighborhood of that pixel. One such statistic is the probability that a particular pixel is an edge pixel.

The first task of the single-image edge detection algorithm is to flag cloudy areas. This task, which can be bypassed if the images are already declouded, operates at the picture and local levels (Cayula and

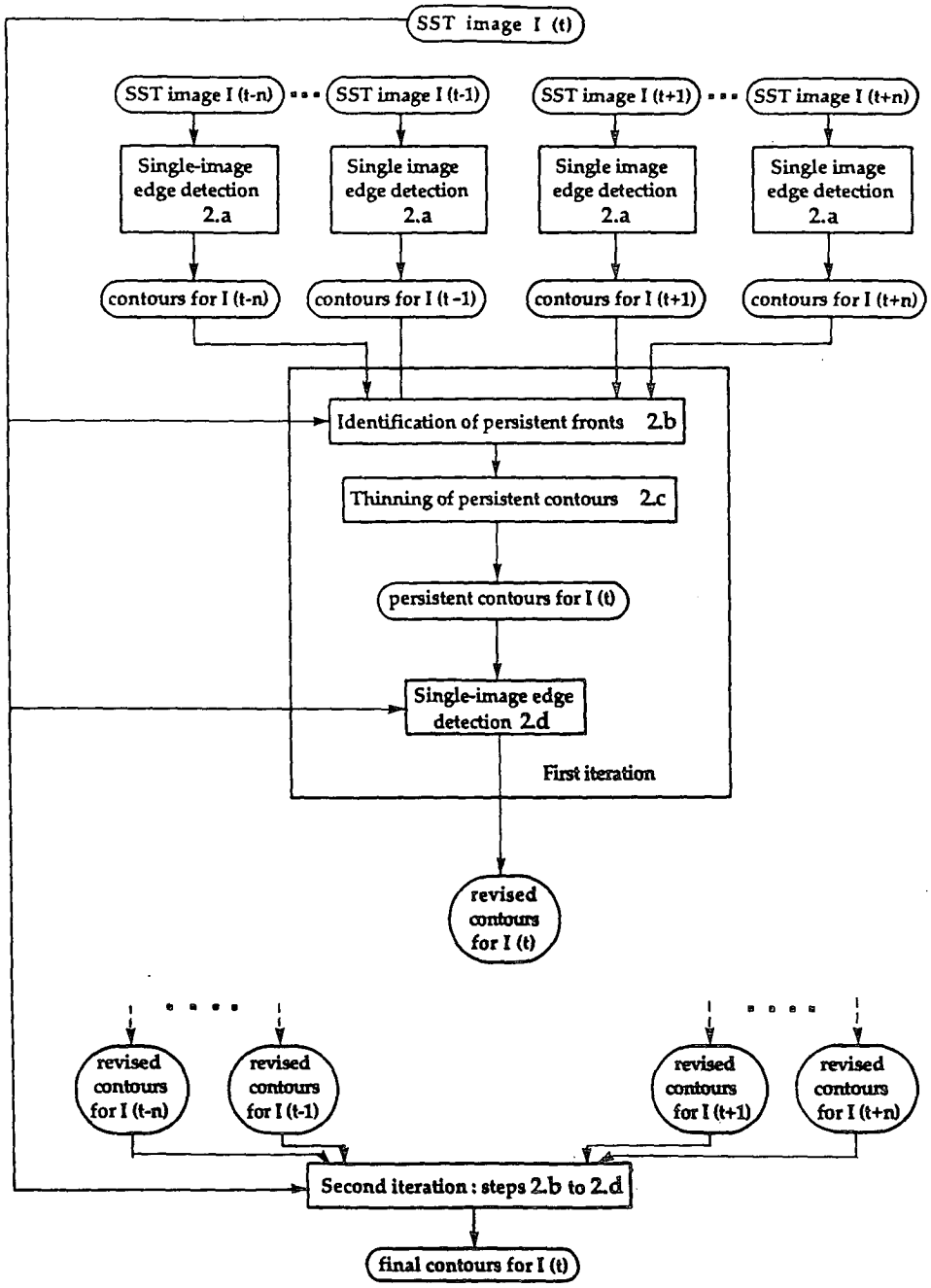


FIG. 1. Flowchart of the algorithm. The number and letter in a box indicates the section of this manuscript in which the step is explained.

Cornillon 1994). The procedure continues at the window level with the formal portion of the edge detection. Using techniques for unsupervised learning, the temperature distribution (histogram) in each window is analyzed to determine the statistical relevance of each possible front. Besides temperature, the algorithm also analyzes the spatial properties of the SST fields to determine whether an object, such as a water mass, forms a compact population. Finally, local operators are introduced to complete the contours that were found at the window level. The output of the single-image algorithm is a list of contours associated with the current image. In that list of contours, the coordinates of every pixel in each contour define the contours. The coordinates also include information about the gradient vector of the SST field at each contour pixel. Figure 3a represents an example of SST imagery. Figure 3b shows the same

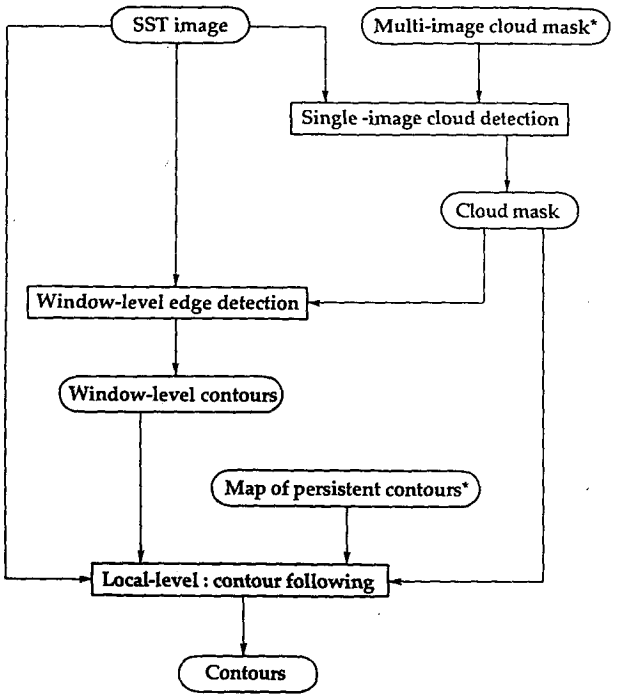


FIG. 2. Flowchart of the single-image edge detection algorithm. The asterisk indicates inputs derived from multi-image processing.

image after processing by the single-image edge detection algorithm.

b. *Identification of the persistent fronts*

This section presents a method to identify, among the fronts in neighboring images that were detected by the single-image algorithm (previous section), those that correspond to features in the SST field of the current image. As shown in Fig. 3c, fronts from neighboring images, within 2.5 days of the current image, show similarities in shape and location to features and fronts in the current image. However, simple translation of the fronts, to match them to features in the current image, is not adequate because oceanographic contours modify their shape with time. To circumvent this problem, contours are repetitively broken into smaller segments before matching.

A contour C of length N is defined by the location coordinates of its pixels,

$$C = \{(x_p, y_p) | p \in [1, N]\}. \quad (1)$$

Each pixel in contour C starts a segment of length λ. The choice of λ depends on such factors as the dynamics of the region under study, image resolution, and the desired robustness of the matching process. A segment S_q of C is defined for $q \in [1, N]$ by

$$S_q = \{(x_p, y_p) | p \in [q, q + \lambda - 1] \cap [1, N]\}. \quad (2)$$

The method used to match a segment S_q of a contour from a neighbor image to the current image relies on a comparison of the gradient fields rather than on the comparison of temperatures. The reason for using the gradient field is that absolute temperatures are not very reliable from image to image, especially if both night and day images are used. To be considered similar, two gradient vectors must have similar magnitudes and similar directions. The function $m(\text{grad}_1, \text{grad}_2)$ measures the similarity between two gradient vectors grad_1 and grad_2. It returns a value ranging from 1 for identical vectors to 0 for vectors that are not comparable. Vectors are considered not to be comparable when the scalar product of the two vectors is lower than 0 or when the magnitude of one vector is infinitely larger than that of the other,

$$m(\text{grad}_1, \text{grad}_2) = \begin{cases} 0, & \text{if } \text{grad}_1 \cdot \text{grad}_2 \leq 0; \\ & \text{otherwise,} \\ \dfrac{\text{grad}_1 \cdot \text{grad}_2}{\|\text{grad}_1\|^2}, & \text{if } \|\text{grad}_1\| > \|\text{grad}_2\|; \\ \dfrac{\text{grad}_1 \cdot \text{grad}_2}{\|\text{grad}_2\|^2}, & \text{if } \|\text{grad}_1\| \leq \|\text{grad}_2\|. \end{cases} \quad (3)$$

The goodness of the match between the segment S_q translated by $\mathbf{t} = (t_x, t_y)$ and the SST features of the current image is estimated as a function of the translation parameters t_x and t_y,

$$M(\mathbf{t}) = \sum_{(x_p, y_p) \in S_q} m[\text{grad}_N(x_p, y_p), \text{grad}_C(x_p + t_x, y_p + t_y)], \quad (4)$$

where $\text{grad}_C(x_p + t_x, y_p + t_y)$ denotes the gradient vector associated with the current image and evaluated at a location $(x_p + t_x, y_p + t_y)$, while $\text{grad}_N(x_p, y_p)$ refers to the gradient vector associated with the neighbor image and evaluated at location (x_p, y_p). For both the current and the neighbor images, the approximation to the gradient vector[1] at a location (i, j) is defined as follows:

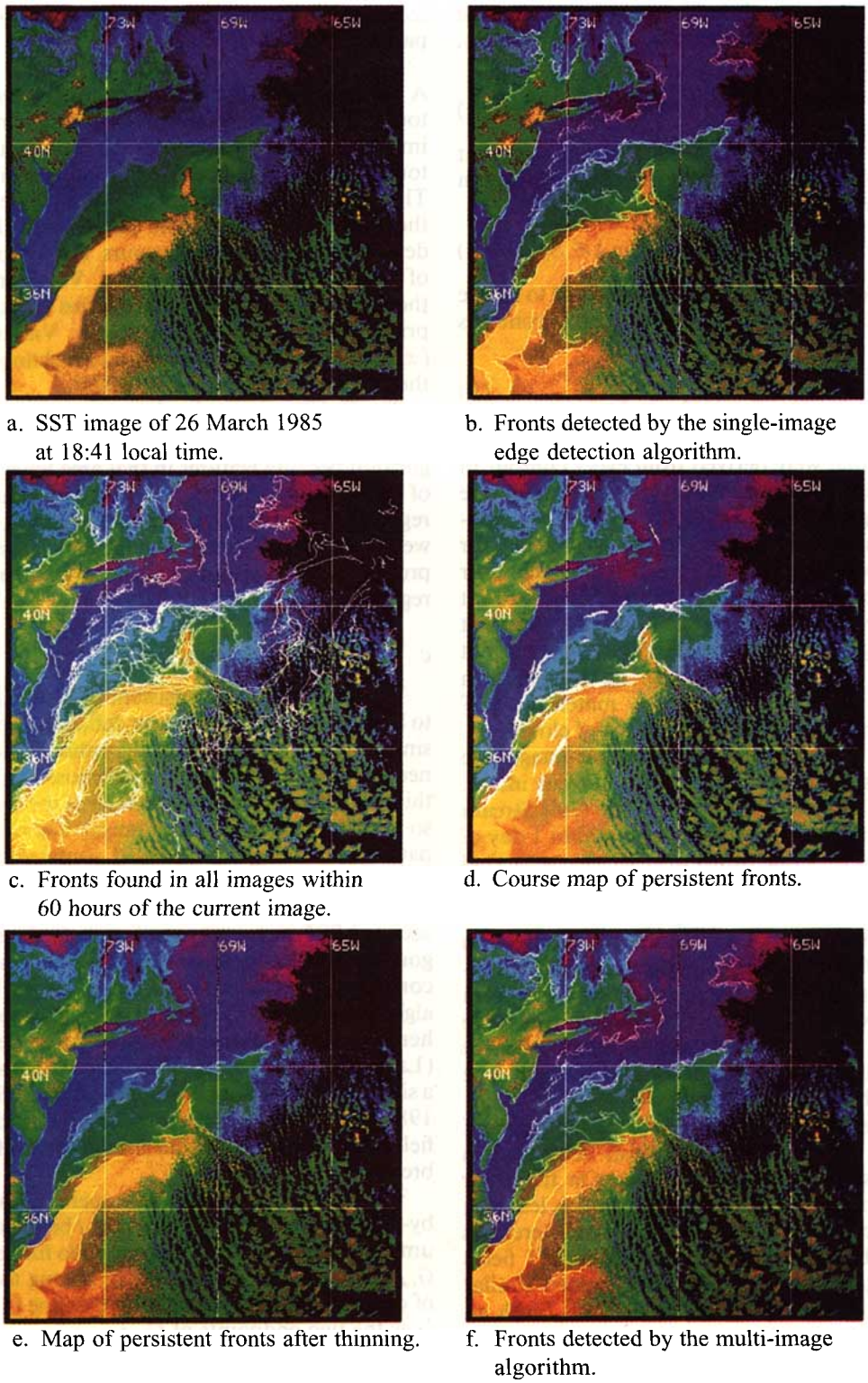
$$\text{grad}(i, j) = \begin{bmatrix} \text{grad}_x(i, j) \\ \text{grad}_y(i, j) \end{bmatrix} = \begin{pmatrix} \tau_{i+1,j} - \tau_{i-1,j} \\ \tau_{i,j+1} - \tau_{i,j-1} \end{pmatrix}, \quad (5)$$

where $\tau_{i+1,j}$, $\tau_{i-1,j}$, $\tau_{i,j+1}$, and $\tau_{i,j-1}$ represent the value of the SST field at locations $(i + 1, j)$, $(i - 1, j)$, $(i, j + 1)$, and $(i, j - 1)$. The vector $\mathbf{t}$ spans a window W,

$$W = [-\delta, \delta] \times [-\delta, \delta], \quad (6)$$

where δ reflects the expected magnitude of frontal displacements for the region and the time interval under

[1] Due to the discrete nature of the data, the step size is measured and used instead of the actual gradient.



a. SST image of 26 March 1985 at 18:41 local time.

b. Fronts detected by the single-image edge detection algorithm.

c. Fronts found in all images within 60 hours of the current image.

d. Course map of persistent fronts.

e. Map of persistent fronts after thinning.

f. Fronts detected by the multi-image algorithm.

FIG. 3. (a) SST image of 1841 LT 26 March 1985. (b) Fronts detected by the single-image edge detection algorithm. (c) Fronts found in all images within 60 h of the current image. (d) Coarse map of persistent fronts. (e) Map of persistent fronts after thinning. (f) Fronts detected by the multi-image algorithm.

consideration. Among all the vectors $\mathbf{t}$ in W, the best match is obtained for the translation vector $\mathbf{T} = (T_x, T_y)$ such that,

$$M(\mathbf{T}) \geq M(\mathbf{t}), \quad \forall \mathbf{t} \in W. \tag{7}$$

The parameters T_x and T_y define the translated segment S'_q, which best corresponds to the gradient features in the current image,

$$S'_q = \{(x_p + T_x, y_p + T_y) | (x_p, y_p) \in S_q\}. \tag{8}$$

The value of $M(\mathbf{T})$ is compared to a threshold to decide whether to include S'_q in the map of persistent contours $\mathcal{P}_C$,

$$M(\mathbf{T}) \geq \mu \Rightarrow S'_q \subset \mathcal{P}_C. \tag{9}$$

The decision process outlined above is repeated for all the segments that were derived from every contour in the neighbor image. When the list of contours for one neighbor image is exhausted, a list of contours associated with another neighbor image is selected for comparison with the current image. All the contour lists corresponding to neighbor images of the current image are used to build the coarse map of persistent fronts $\mathcal{P}_C$ (Fig. 3d). Note that the edge of the eddy at 39°N, 69°W, which appears in Fig. 3c, disappears in Fig. 3d because it does not properly match the SST fields in that region due to the presence of clouds.

Parameters are set as follows. Neighbor images are all the images within 2.5 days of the current image. This reflects the observation that in the Gulf Stream region most matches occur for images within 1 day of the current image, while fronts from images separated by more than 3 days add little information to fronts in the current image. Increasing the maximum time separation from 2.5 to 3.5 days results in the detection of less than 1% additional edge pixels and a 20% increase in computing time for our dataset. Away from the western boundary currents, features move more slowly and the algorithm would more noticeably benefit from a larger temporal window. This is especially true in regions with significant cloud cover.

Two factors determine the choice of the segment length λ. On one hand, the algorithm becomes less robust to noise as λ becomes smaller. On the other hand, with longer segments, the algorithm is less able to take into account contour deformations. Here, λ is set to 20 pixels as tests with images of the SST fields off Cape Hatteras (1-km resolution) yielded similar results for segment lengths of 10–40 pixels (with an appropriately chosen μ).

The window size parameter δ is set to 10 km to reflect the expected displacement of Gulf Stream meanders in 1 day. But other values could also be used. With δ equal to 15 km, one can observe a 5% increase in the number of detected edges. However, this increase is accompanied by a doubling of the processing time. A 7-km δ results in a 5% decrease of the number of detected edge pixels but half the processing time compared to δ set to 10 km.

The selection of μ is a trade-off between two factors. A lower μ increases the risk of wrongly matching contours (from neighbor images) to features (in the current image). A higher μ increases the probability that contours that actually correspond to features will be missed. The value of μ also depends on the segment length λ, the amount of noise in the images, and the expected deformation rate of the contours. For a segment length of 20 pixels, $\mu = 10$ was selected. Visual inspection of the results empirically confirms this choice as appropriate for the Gulf Stream region. Variations of ± 1 ($\pm 10\%$) in the value of μ produce a change of $\mp 6\%$ in the number of detected edge pixels.

The Gulf Stream region is expected to place the most difficult demands on the multi-image parts of the algorithm because features in that area have higher rates of displacement and deformation than features in other regions of the oceans. As the values for λ, δ, and μ work well for the Gulf Stream region, they represent an appropriate choice, although a conservative one, for other regions of the oceans.

c. *Thinning of persistent fronts*

The coarse maps of persistent fronts are not suitable to be processed by the contour-following portion of the single-image edge detection algorithm. Thick contours need to be thinned. The method chosen to accomplish this task relies on the gradient field of the current image so that the output of the thinning algorithm is compatible with the gradient-based contour-following algorithm (Cayula and Cornillon 1992).

The coarse map of persistent contours and the associated SST image serve as inputs to the thinning algorithm. The coarse connected contours can often become disconnected in the thinning process because the algorithm relies on gradient information, which is inherently noisy. Although more sophisticated solutions (Lam et al. 1992) exist to obtain connected contours, a simple solution was chosen. Median filtering (Huang 1981) the image before computing the gradient vector field was sufficient to avoid the problem of artificially breaking up contours.

The thinning algorithm first operates on a column-by-column basis. For example, let i be the current column number. Column i is scanned to find every group $G_{i,k}$ of consecutive pixels, which belong to the set $\mathcal{P}_C$ of coarse persistent contours. We define (i, k) and (i, l) as the two endpoints of $G_{i,k}$:

$$(i,j) \in G_{i,k} \Leftrightarrow \begin{cases} (i, k-1) \notin \mathcal{P}_C; \\ (i, l+1) \notin \mathcal{P}_C; \\ \forall j \in [k, l], \quad (i,j) \in \mathcal{P}_C. \end{cases} \tag{10}$$

Thinning is achieved by selecting, at most, a single pixel from $G_{i,k}$. The algorithm examines the value of

$\text{grad}_y(i,j)$ in (5) for each pixel (i,j) in $G_{i,k}$ to determine which pixel to include in the map $\mathcal{P}_T$ of thinned persistent contours:

$$(i, m) \in \mathcal{P}_T \Leftrightarrow \begin{cases} (i, m) \in G_{i,k}; \\ \forall (i,j) \in G_{i,k}, \ |\text{grad}_y(i, m)| \geq |\text{grad}_y(i,j)|; \\ |\text{grad}_y(i, m)| \geq \epsilon. \end{cases} \tag{11}$$

The threshold ϵ helps to eliminate small artifacts produced by the thinning algorithm and image quantization. The value of the threshold has to be chosen to reflect the level of quantization and also the amount of noise, which is present in the images. Here, ϵ is defined as 2 digital counts, that is, a 0.25°C step over a 2-pixel spacing (4 km for the development dataset and 2 km for the validation dataset).

After all columns have been analyzed, the algorithm continues with row-by-row processing. The method followed for row processing is similar to that used for column processing except that, for row j, (10) becomes

$$(i,j) \in G'_{r,j} \Leftrightarrow \begin{cases} (r-1, j) \notin \mathcal{P}_C; \\ (s+1, j) \notin \mathcal{P}_C; \\ \forall i \in [r, s], \quad (i,j) \in \mathcal{P}_C, \end{cases} \tag{12}$$

and (11) changes to

$$(m, j) \in \mathcal{P}_T \Leftrightarrow \begin{cases} (m,j) \in G'_{r,j}; \\ \forall (i,j) \in G'_{r,j}, \ |\text{grad}_x(m,j)| \geq |\text{grad}_x(i,j)|; \\ |\text{grad}_x(m,j)| \geq \epsilon. \end{cases} \tag{13}$$

The output of the thinning algorithm is shown in Fig. 3e.

d. Single-image edge detection—Subsequent passes

Each map of persistent contours is used to help detect fronts that were missed by the previous application of the single-image edge detection algorithm either because they were too weak or because of cloud cover. To this end, the single-image algorithm has been modified to accept both an image and its associated map of persistent contours as inputs. In the modified version of the algorithm, contours flagged in the map of persistent contours are added to those detected at the window level by the original single-image edge detection algorithm. The algorithm then proceeds with local-level processing to produce a revised list of contours. In this manner, persistent contours serve as seeds for the single-image algorithm to detect new contours that may actually be longer than the persistent contours from which they are derived. Figure 3f shows the SST field of Fig. 3a with the contours from the revised list overlaid in white. Note the detection of the weak front at 36°N, 74°W and of the cloud-contaminated front at about 38°N, 69°W. Both fronts were originally missed by the single-image edge detection algorithm of section 2a when it was applied to the image shown in Fig. 3a (Fig. 3b). Few pixels were added to the persistent front passing through 38°N, 69°W by the single-image edge detection algorithm, while the persistent front passing through 36°N, 74°W was significantly extended by the contour-following portion of the algorithm.

3. Validation

Before conducting the evaluation, it was necessary to select the northern edge of the Gulf Stream from among all the other detected edges. This postprocessing step is either carried out by an expert or, automatically, with the Gulf Stream selection algorithm.

The automatic selection method, which is more completely described in Cayula (1993), follows a three-step process. First, for all the images in the sequence, a crude search area is defined from past observations of the extreme position of the Gulf Stream northern edge. Then, for each image, the location of the warmest pixels along the Gulf Stream helps to further limit the search area, the front must be shoreward of the line of warmest pixels. Finally, within the search area, an edge pixel is selected based on two factors: the direction of the associated gradient vector at that location and the SST values of the neighboring pixels.

To evaluate its performance, the multi-image edge detection algorithm was applied to a set of 98 satellite-derived SST images that were used in previous comparative studies (Cayula and Cornillon 1992; Cornillon and Watts 1987). In those studies, the location of the Gulf Stream northern edge off Cape Hatteras (34°–39°N, 70°–76°W), as determined by various methods from satellite imagery, was compared with the location of the Gulf Stream determined from inverted echo sounders (IES) (Chaplin and Watts 1984) moored on the sea floor. The position of the Gulf Stream that is determined from inverted echo sounders corresponds to the surface projection of the 15°C isotherm at 200 m T_{15}, one of the generally accepted definitions of the northern edge. During the period covered by the set of 98 images, three inverted echo sounder lines were available. Hence, a maximum of 294 satellite–IES comparisons were possible. In fact, an expert, in a careful analysis of the images, was only able to find 226 points of comparison. The satellite data corresponding to the remaining 68 possible comparisons were simply too cloudy.

Three statistics are computed to analyze the offset δ between the locations of the Gulf Stream northern edge detected in satellite-derived imagery and of the corresponding front obtained from in situ data. This offset

is negative when the SST-located front is south (seaward) of T_{15} and positive otherwise. The first statistic is the mean offset between the SST-located front and T_{15},

$$m_\perp = \frac{1}{N} \sum \delta, \tag{14}$$

where N is the number of observations for which both T_{15} and the corresponding satellite-derived location of the Gulf Stream northern edge have been determined. The subscript ($\perp$) in (14) indicates that δ is the separation measured perpendicular to the satellite front. Because isotherms intersecting the sea surface are not vertical and because the surface projection of T_{15} is being compared with the surface front, a nonzero-mean offset is expected. The scatter about the mean offset is a better measure of how well the edge detected from satellite-derived imagery estimates T_{15},

$$\sigma_\perp^2 = \frac{1}{(N-1)} \sum (\delta - m_\perp)^2. \tag{15}$$

Further insight into the quality of the SST edge as an estimate of T_{15} is obtained by studying the skewness of the distribution of the offsets,

$$\alpha_{3\perp} = \frac{1}{(N-1)\sigma_\perp^3} \sum (\delta - m_\perp)^3. \tag{16}$$

Table 1 summarizes the results obtained by the (original) single-image algorithm (Cayula and Cornillon 1992), the single-image algorithm with multi-image cloud detection (MICD), the multi-image algorithm, and the analyst. The last line of Table 1 shows the results achieved when an analyst selected the northern edge of the Gulf Stream among the edges that were found by the multi-image algorithm. The computation of the statistics for the automatic methods uses all edges that are considered reliable by the Gulf Stream selection algorithm (Cayula 1993; Cayula and Cornillon 1992). Similarly, the statistics for the subjective method are computed for all detected edges, which satisfy the minimum acceptable quality level for a reliable detection as determined by the analyst (Cornillon and Watts 1987). In Table 1, the variations of $\sigma_\perp$ are not considered statistically significant. For sets of 200 normally distributed samples with a standard deviation of 15.4 km, Monte Carlo simulation indicates that variations of less than ±0.7 km about the mean value of the standard deviation are not statistically significant at the 0.95 level of significance. By contrast, the scatter $\sigma_\perp$ for SST fronts selected by other automated methods (gradient, skew, variance, and maximum temperature) is significantly higher, $\sigma_\perp > 18$ km (Cornillon and Watts 1987), than those discussed here.

The most striking figure in Table 1 is the low value of N achieved by the (original) single-image algorithm compared to either the multi-image algorithm or the analyst. The multi-image edge detection algorithm is

TABLE 1. Results for various edge detection methods.

Method	N	$m_\perp$	$\sigma_\perp$	$\alpha_{3\perp}$
Single-image algorithm	123	−13.22	15.79	0.23
Single-image algorithm w/MICD	157	−15.20	15.16	−0.28
Multi-image algorithm	192	−12.78	15.06	−0.05
Subjective analysis	226	−12.53	14.91	−0.10
Multi-image algorithm w/manual selection	223	−12.52	13.87	−0.04

able to find nearly 60% more fronts in the satellite data that correspond to available T_{15} locations than the single-image edge detection algorithm. This result is due to three factors:

1) The multi-image edge detection algorithm finds a higher number of actual fronts than the single-image edge detection algorithm.

2) The additional fronts detected by the multi-image algorithm are mostly good fronts because

- A false front detected in one image is unlikely to match an actual oceanographic feature in another image,
- A valid SST front detected in one image is unlikely to match a cloud-induced artifact.

3) The multi-image cloud detection by identifying additional clouds leads to the detection of fewer false fronts.

Although indirect, the effect of the second and third points on N accounts for about 50% of the increase. These two points mean that the fronts detected by the multi-image algorithm are more reliable than those detected by the single-image algorithm. The Gulf Stream selection algorithm developed for the multi-image algorithm takes advantage of this increased reliability to reduce, compared to the selection algorithm developed for the single-image algorithm, the number of valid SST fronts that are incorrectly rejected. It is worth noting that the search for the Gulf Stream northern edge is conducted from south to north and terminated when the front is found. As a consequence, cloud-induced fronts south of the northern edge are the ones that most affect the selection process. The positive value for $\alpha_{3\perp}$ for the single-image edge detection is an indication that false fronts are interfering with the selection process.

To isolate the effect of the cloud detection and Gulf Stream selection process, the single-image edge detection algorithm was tested with the same multi-image cloud detection and edge selection algorithms used by the multi-image algorithm. Table 1 shows that with all other conditions being the same, the multi-image algorithm detects 22% more fronts than the single-image edge detection algorithm. The values of $m_\perp$ and $\alpha_{3\perp}$ indicate that the single-image algorithm tends to miss the weaker southernmost edge, which produces the best estimate for the location of T_{15}.

The next two lines of Table 1 show that the multi-image algorithm and the analyst obtain very comparable results. The multi-image algorithm detects about 85% of the fronts detected by the analyst. However, it is possible that better results could have been obtained if the sequence of images used to evaluate the algorithm had been more complete (Cayula 1993). For example, with the temporal window set to 5 days, each image in the set used to develop the multi-image algorithm (Cayula 1993) had an average of eight neighbor images versus only four for images in the evaluation set. When comparing the multi-image algorithm to the single-image version, an estimated 40% increase in the number of actual edge pixels detected was observed for the development set instead of 24% for the evaluation set. The last line of Table 1 indicates that the Gulf Stream northern edge selection algorithm is also a problem. When the fronts detected by the multi-image algorithm are manually selected, the multi-image algorithm performs significantly better than the expert: the number of good coincidences found is essentially the same but the standard deviation is significantly smaller.

On the evaluation set of 98 images, the processing time for the complete algorithm was between 20 and 30 min per image on a DECstation 5000/200 ($\approx$20 MIPS). For each application, the single-image analysis needed about 30 s per image, the persistent contour identification, 10 min, and the multi-image cloud detection, 5 min.

4. Summary

This manuscript presents a framework for using information found in a sequence of images to detect edges in one image. Although it uses simple techniques, the multi-image algorithm is shown to significantly improve the results obtained by a single-image edge detection algorithm. In particular, the use of information from other images allows the detection of weaker or corrupted fronts with little increase in the number of false contours. On the test set of images, the multi-image edge detection algorithm, associated with the multi-image cloud detection, performs nearly as well as the analyst. A more complete set of images, optimized parameters, and a more sophisticated Gulf Stream selection algorithm, perhaps using a multi-image approach, would likely improve the present results.

Acknowledgments. This research was performed with support from the National Aeronautics and Space Administration (NAGW-3009) and from the State of Rhode Island and Providence Plantations through salary support to P. Cornillon. The satellite data processing software used to preprocess the imagery was developed by R. Evans, O. Brown, J. Brown, and A. Li of the University of Miami. Their continued support is greatly appreciated.

REFERENCES

Cayula, J.-F., 1993: Automatic front detection using a sequence of SST images. Ph.D. dissertation, Department of Electrical Engineering, University of Rhode Island, 67 pp.

——, and P. Cornillon, 1992: Edge detection algorithm for SST images. *J. Atmos. Oceanic Technol.*, **9,** 67–80.

——, and ——, 1994: Cloud detection from a sequence of SST images. *Remote Sens. Environ.*, submitted.

——, ——, R. J. Holyer, and S. H. Peckinpaugh, 1991: Comparative study of two recent algorithms designed to process sea-surface temperature fields. *IEEE Trans. on Geosci. and Remote-Sens.*, **GE-29,** 175–177.

Chaplin, G., and D. R. Watts, 1984: Inverted echo sounder development. *Ocean '84 Conf. Record,* **1,** 249–253.

Cornillon, P., and D. R. Watts, 1987: Satellite thermal infrared and inverted echo sounder determination of the Gulf Stream northern edge. *J. Atmos. Oceanic Technol.*, **4,** 712–723.

——, C. Gilman, L. Stramma, O. Brown, R. Evans, and J. Brown, 1987: Processing and analysis of large volume of satellite-derived thermal infrared data. *J. Geophys. Res.*, **92,** 12 993–13 002.

Gerson, D. J., and P. Gaborski, 1977: Pattern analysis for automatic location of ocean fronts in digital satellite imagery. U.S. Naval Oceanographic Office, TN 3700-65-77, 67 pp.

Gilman, C., and L. Rothstein, 1992: Thermal and potential anomalies in the Gulf Stream. *J. Geophys. Res.*, submitted.

Holyer, R. J., and S. H. Peckinpaugh, 1989: Edge detection applied to satellite imagery of the oceans. *IEEE Trans. Geosci. and Remote-Sens.*, **GE-27,** 46–56.

Huang, T. S., Ed., 1981: *Two-Dimensional Digital Signal Processing II: Transforms and Median filters.* Springer-Verlag, 222 pp.

Kahn, P., 1987: Determining temporal persistence and consistent edge motion from natural images. Tech. Rep. University of Massachusetts, Amherst, COINS 87–56, 48 pp.

Lam, L., S.-W. Lee, and C. Suen, 1992: Thinning methodologies—A comprehensive survey. *IEEE Trans. on Pattern Anal. and Machine Intell.*, **PAMI-14,** 869–885.

McClain, E. P., W. G. Pichel, C. C. Walton, Z. Ahmad, and J. Sutton, 1983: Multi-channel improvements to satellite-derived global sea surface temperatures. *Adv. Space Res.*, **2,** 43–47.

Podestà, G., J. Browder, and J. Hoey, 1993: Exploring the association between swordfish catch rates and thermal fronts in the western North Atlantic. *Continental Shelf Res.*, **13,** 253–277.

Pratt, L. J., J. Earles, P. Cornillon, and J.-F. Cayula, 1991: The nonlinear behavior of varicose disturbances in a simple model of the Gulf Stream. *Deep-Sea Res.*, **38,** S591–S622.

Schwalb, A., 1978: The TIROS-N/NOAA-G satellite series. NOAA Tech. Memo. NESS 95, National Environmental Satellite Service, NOAA, U.S. Dept. of Commerce, 77 pp.

Snyder, M. A., 1989: The precision of 3-D parameters in correspondence-based techniques: The case of uniform translational motion of a rigid environment. *IEEE Trans. on Pattern Anal. and Machine Intell.*, **PAMI-11,** 523–528.