FY99 Progress Report:
Plume Detection and Tracking in Video Data[1]

Won-Ha Kim
and
Christopher Brislawn

October 21, 1999

Los Alamos National Laboratory

Computer Research/Applications (CIC-3)

Mail Stop B265

Los Alamos, NM 87545

Email: wonha@lanl.gov, brislawn@lanl.gov

[1]Los Alamos tech report number LAUR-00-15

Contents

I. Introduction

Segmentation and identification of moving objects are crucially important tasks for video analysis applications such as robot vision, airborne surveillance and content-based video coding [1]. The purpose of segmentation is to automatically detect object movements and track the object regions. The purpose of identification is to extract the target object features and pick objects based on their the features.

So far, there has been extensive research on segmenting and identifying rigid objects or model-based deformable objects [2] [3] [4]. However, the segmentation and tracking of gaseous objects is rarely studied. In this work, we develop a system for tracking gaseous objects via image segmentation and identification. A good example of such objects is a smoke plume. Plume detection is a specific application for airborne surveillance. Difficulties in tracking gaseous objects arise from the irregularity of objects' shapes and motion behavior and the low resolution of the image. Since the intensity of a gaseous object is variable at the boundary, the object's boundaries can not be well defined. Furthermore, since the motion of gaseous objects depends on air turbulence, it is hard to decide time order of plume sequence and region of a same plume may be spatially disconnected. Therefore, modeling such an object's shape and motion behavior is very intractable. In many practical image sequences, object images are noisy and object sizes are small compared to the image resolution. Thus the texture patterns on objects are indiscernible, which makes object identification more difficult.

Good motion estimation is a key ingredient in motion based segmentation. Thus, the motion estimation and segmentation must be address simultaneously for best results. In relation to motion-based object segmentation, motion estimation/analysis methods are mainly classified as either *Parametric motion method* or *Non-Parametric method* [5].

One parametric method is the affine flow model. Under the orthographic projection of a 3-D rigid object to a 2-D object, the method represents an object's motion field as a state space and fits parameters on the state space to the motion of the object [6]. The affine model represents the spatial-temporal variation of the motion field in terms of affine transforms. More affine parameters describe more sophisticated motions. Two affine parameters represent the simple model of a 2-D rigid object moving parallel to the image plane. The 3-D rigid body motion of planar objects leads to six affine parameters under parallel projection [2] and to eight parameters under prospective projection [7]. The affine flow model is attractive because it is relatively simple, yet is able to model various motions such as translation, zooming, and rotation of rigid objects. However, the method is not applicable to non-rigid objects.

As a non-parametric method, block matching is extensively used in most video codecs such as H.263, and MPEG [8] [9]. Block matching assumes a simple motion model. It checks fixed-size blocks moving translationally in the image plane. The method divides the frame at time t into blocks and then searches for the corresponding blocks at $(t-1)$. While the block matching method is simple and not very vulnerable to image noise, the idea of translational motion of fixed size blocks cannot properly describe the detail region of moving objects. Thus, the block matching method is suitable for estimating only simple types of motion.

Another non-parametric method is based on the optical flow equation [10] providing an estimate of the optical flow field in terms of spatial-temporal image intensity gradients. This method usually produces a dense motion field for each image frame. "Dense motion field" describes a motion field with independent optical flow vectors for each pixel. The models project continuous 3-D motions to 2-D motions in image space, and extract homogeneous motion regions, relying on motion models and spatial-temporal image gradients. The dense motion field method is ill-conditioned since different motion field vectors are associated with the same data. So, it requires a regularization by imposing smoothness or uniformity constraints on the local motion field. In these methods, the Markov Random field (MRF) is often used to model images, and the smoothness constraint is expressed as an energy function of the associate Gibbs distribution [11] [12]. They set up an objective function as a sum of the energy of motion estimation error and the weight smoothness constraint. Based on the maximum a posterior (MAP) criterion, motion is then estimated by searching the motion field minimizing the objective function [13] [14]. Drawbacks of this model are vulnerability to noise, heavy complexity due to simultaneous computations of large unknown variables and difficulity for depicting specific shape information [4]. A motion model incorporating both parametric and dense motion model is also employed to represent global motion with parametric components and describe local motion with dense motion vector field [5].

Since it is not possible to model gaseous objects, the affine model cannot be used to estimate gaseous objects' motion. The object images are corrupted with noise and many applications with gaseous object tracking require less complexity. So, the optical flow method is also inappropriate for detecting the gaseous objects. Therefore, we use the block matching method to detect motions of gaseous objects. To overcome the insufficient description of object regions in block matching, we further applied some post-processing with morphological filtering [15] to the block based regions.

In moving object segmentation schemes, the object regions are linked in time, meaning that each moving object must be tracked temporally. The gaseous objects tend to spread to lower air density regions. So, unlike rigid object motion, the gaseous objects vary in intensity within a single object region, and regions of the same gaseous object may even be disconnected. Thus, traditional moving object segmentation methods such as the model-based method [16] [17], the

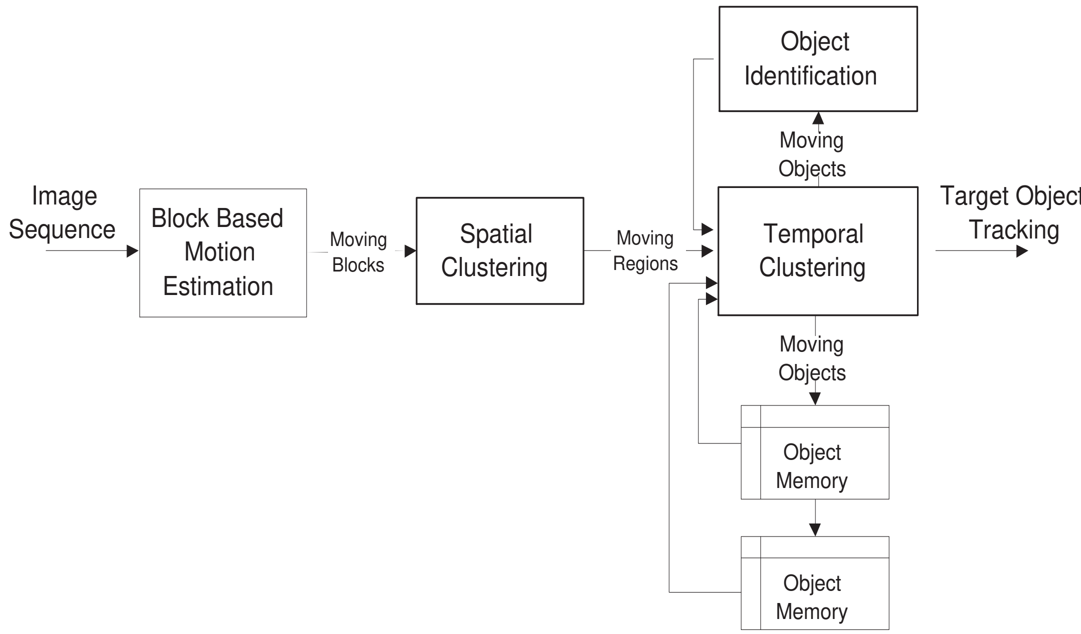


Fig. 1. Overview of a gaseous object tracking system.

region-based method [18], or the intensity-based methods [19] are not well-suited for gaseous object segmentation. In this work, we propose a novel temporal clustering method based on tracking object paths. The proposed method records the previous object regions to track the paths of spreading objects and clusters the objects on a path into a single object. The method is especially good at joining disconnected regions of gaseous objects. In addition, the proposed spatial-temporal segmentation method fits the current content-based coding standard of MPEG-4 [3].

Tracking an object requires identifying the target object from amongst the moving objects. A typical feature used for identifying a non-rigid object could be a texture pattern [20]. However, in actual circumstances such as airborne surveillance, noise and low resolution of object images hide distinct texture patterns. Therefore, as object features, we use boundary direction pattern, shape and energy distribution on frequency bands instead of texture information.

Figure 1 illustrates the overall view of the proposed gaseous object tracking system. The motion block matching method detects moving blocks. The spatial clustering clusters the blocks into objects. The temporal clustering tracks motion behavior of the spatially clustered objects and clusters the objects following the same motion path into the same object. The results of temporal clustering are saved as object memories. The object memories provide object motion behaviors for the next temporal clustering process. At a given frame time, the object identification process extracts features of the objects obtained after temporal clustering and targets the objects that match with the desired object features.

II. Moving Object Segmentation

In this section, we develop a novel moving object segmentation method that works well especially for gaseous objects. The method we employ is a hierarchically structured procedure. In the spatial domain, we first detect moving blocks and then merge the moving blocks based on similarities so as to produce moving regions. In the temporal domain, we track the paths of moving regions and cluster the moving regions into moving objects.

A. Block-Matching Motion Estimation

In surveillance images, the object sizes are usually small and the object images are corrupted with noise. So, it is difficult to detect pixel-to-pixel motion. Furthermore, the pixel-to-pixel motion detection inherently requires a high computation load. Thus, an optical flow method is inappropriate to be used in surveillance. The parametric method such as the affine model is also not applicable to gaseous object motion detection since the affine model is useful for only rigid object motion. So, we use block-matching motion estimation which is used in current MPEG coding standards.

We slice an image plane into blocks of size $n \times n$ pixels. The value of n depends on object size and resolution. We adopt the three step search method for motion displacement search [8] [9]. At each sequential step, the search ranges are 3-pixel, 2-pixel and 1-pixel. We use mean square error (MSE) as a matching criteria since MSE is more convenient for determining a proper moving block threshold. Suppose that the center of a block B_i is (x, y) at t and moves from $(x + \triangle_x, y + \triangle_y)$ at $(t + 1)$. Let I_t be the image frame at time t. So, $I_t(x, y)$ is the image pixel intensity at (x, y). The block-matching MSE and the block displacement are then calculated as

Block-Matching MSE :

$$MSE(B_i) = \frac{1}{n^2} \sum_{(i,j) \in B_i} |I_{t+1}(x, y) - I_t(x + \triangle_x + i, y + \triangle_y + j)|^2$$

Block displacement : $\quad \Delta(B_i) = \sqrt{\triangle_x^2 + \triangle_y^2}\ .$

The moving blocks detected by motion estimation may include irrelevant detected blocks due to noise or noise-like motion such as motion of leaves on trees. To distinguish such noisy motion blocks, the block-matching MSE of real moving blocks must be larger than a certain value, that is, a threshold. So, the decision of the threshold value is crucial in motion estimation performance. We set up the block-matching MSE threshold through a significance test [21]. At (x, y) where no pixel movement occurs, $d(x, y) = I_t(x, y) - I_{t+1}(x, y)$ is a camera noise. It is well-known that camera noise follows a zero mean Gaussian distribution with a variance σ^2 equal to twice the variance of the camera noise [21]. So, if we let H_0 be the null hypothesis that there is no change at (x,y), the probability density function (*pdf*) of the camera noise is

$$p(d(x,y)|H_0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{d^2(x,y)}{2\sigma^2}\} \ .$$

We can define T as the normalized block-matching MSE where

$$T = \frac{1}{\sigma^2 \cdot n^2} \sum_{(x,y)\in B_i} d^2(x,y)$$

where B_i is a block of size $n \times n$ pixels, and σ^2 is twice the camera noise variance.

Assuming that camera noise $d(x,y)$ is statistically independent, the variable T follows a χ^2 distribution with n^2 degrees of freedom. We choose the threshold t_α through the significance test. A significance level α corresponding to t_α is

$$\alpha = p_{\chi^2}(T > t_\alpha | H_0)$$

where $p_{\chi^2}(T)$ is χ^2 distribution.
Thus, α is the probability of type I. In other words, the significant level α can be considered as the probability of detecting a block B_i as a moving block although the block does not move. For 16×16 blocks, we set t_α to be 1, which corresponds to 90% confidence.

Figure 3(a) shows the result of a plume motion detection with 16×16 blocks.

B. Moving Region Recovery via Spatial Clustering

In the motion estimation process, an image plane was divided into blocks. So, a moving region is set of moving blocks. We need to merge similar moving blocks to recover moving regions, which means motion block clustering in the spatial domain. We perform a non-supervised clustering method. For clustering features, we use block connectivity, normalized difference of block-matching MSE and normalized difference of block displacement. We do not use intensity since gaseous object intensity varies even within a single object.

We say two blocks to be neighbors if they share an edge or vertex, that is, they are in 8-connectivity. Given two moving blocks B_i, B_j, we define similarity measures as follows;

$$\text{Connectivity}: Con(B_i,B_j) = \begin{cases} 1 & \text{if blocks } B_i, B_j \text{ are neighbors.} \\ \infty & \text{if blocks } B_i, B_j \text{ are } \textit{not} \text{ neighbors.} \end{cases}$$

$$\text{Normalized MSE difference}: MSE(B_i,B_j) = \left| \frac{MSE(B_i) - MSE(B_j)}{MSE(B_i) + MSE(B_j)} \right|$$

$$\text{Normalized Displacement difference}: \Delta(B_i,B_j) = \left| \frac{\Delta(B_i) - \Delta(B_j)}{\Delta(B_i) + \Delta(B_j)} \right| \ .$$

Then, the similarity distance between moving blocks B_i, B_j is obtained as

$$D_c(B_i, B_j) \equiv Con(B_i, B_j) \cdot \{\theta \cdot MSE(B_i, B_j) + (1-\theta) \cdot \Delta(B_i, B_j)\} \tag{1}$$

where the weight factor θ is 0.3 for our plume data. This value was determined by training.

Blocks B_i, B_j are merged into the same object region if the similarity distance is less than a certain threshold value. Otherwise, B_i, B_j are regarded as different object blocks. In our plume data, we set the threshold distance to be 0.5 by experimental training. Therefore, if B_i, B_j are connected and $D_c(B_i, B_j) < 0.5$, B_i, B_j are merged to a object region.

We perform clustering with the one-path region-growing method. At first, each cluster is initialized as a single moving block. If the first moving block and the next moving block are to be merged, those blocks are merged to the first cluster. Otherwise, the next block remains as a cluster. This process is continued for each new moving block until all moving blocks are tested. For the formal description of spatial clustering algorithm, we use the following definitions and notation.

Definitions and Notation

- $o(i)$: Index of ith element in a set. For example, if a set $G = \{g_2, g_4, g_6, g_9\}$, $g_{o(1)} = g_2$, $g_{o(2)} = g_4$, $g_{o(3)} = g_6$ and $g_{o(4)} = g_9$.
- $S(k)$: kth element of a set S. If $S = \{2, 3, 5, 6\}$, $S(1) = 2, S(2) = 3, S(3) = 5$ and $S(4) = 6$.
- B_i : A moving block.
- **MB** : Set of all moving blocks. So, if blocks $B_i, B_j, \ldots B_t$ moved, $\mathbf{MB} = \{B_i, B_j, \ldots B_t\}$.
- R_i : Set of moving blocks in the region indexed by i. For example, if blocks $B_h, B_l, \ldots B_t$ are clustered to the i-indexed region, $R_i = \{B_h, B_l, \ldots B_t\}$.
- **REG** : Set of moving regions. If the number of moving region is l, that is, $|\mathbf{REG}| = l$, then $\mathbf{REG} = \{R_1, R_2, \ldots R_l\}$.
- $\mathbf{TouchBl}(R_i, B_k)$: Function finding a block in R_i touching a block B_k where B_k does not belong to R_i. So, $\mathbf{TouchBk}(R_i, B_k) = B_t$ implies that $B_t \in R_i$ and $Con(B_t, B_k) = 1$.

Based on these definitions and notation, we describe the spatial clustering algorithm;

Moving Block Clustering Algorithm

/* Initialization */
$\mathbf{REG} = \mathbf{MB}$; /* So, $R_i = B_i$. */
$i = 1$; /* Initialize object index */
While ($\mathbf{MB} \neq \emptyset$)
    For k=1 to $|\mathbf{MB}|$
    /* Check if kth block in **MB** touches ith region in **REG** */

If ($\mathbf{TouchBl}(R_{o(i)}, B_{o(k)}) \neq \emptyset$)

Let $B_p = \mathbf{TouchBl}(R_{o(i)}, B_{o(k)})$.

/* If the kth block is clustered to the ith region, expand ith region to cover the block, and register the index of the block. So, A_i is the set of index for blocks touching the ith region. */

If ($D_c(B_p, B_{o(k)}) < t_c$), then $R_{o(i)} = R_{o(i)} \cup \{B_p\}$; $A_i = A_i \cup \{o(k)\}$; End

End

End

/* Remove blocks clustered to an object from **MB** . */

$$\mathbf{MB} = \mathbf{MB} \setminus \left(\bigcup_{j=1}^{|A_i|} B_{A_i(j)} \right);$$

/* Remove regions covered by the ith region from **REG**, in order that regions in **REG** are not duplicated. */

For h=1 to $|\mathbf{REG}|$

If ($R_{o(h)} \subseteq R_{o(i)}$), then $\mathbf{REG} = \mathbf{REG} \setminus \{R_{o(h)}\}$; End

End

/* Update **REG** to include updated ith region. */

$\mathbf{REG} = \mathbf{REG} \cup \{R_{o(i)}\}$;

/* Clustering for $(i+1)th$ region */

$i = i + 1$;

End

Moving region : **REG**

Figure 3(b) illustrates the spatial clustering result. In the figure, moving blocks are clustered into two regions.

C. Moving Object Clustering in the Temporal Domain

One difficulty in segmenting gaseous objects is that regions of a gaseous object may be disconnected. For example, as seen in Figure 3(b), a plume may spread to separated regions. Such separated regions should be observed as a single object region. To overcome this phenomenon, we exploit inter-frame motion tracking. Although the regions formed by an object might not be connected in a single frame, the motion trajectories of such objects will intersect. This means that the separated objects can be connected in the temporal domain.

We perform second order temporal clustering. The object regions of the previous two frames are stored for tracking object motion paths. The motion trajectory of a region in the current frame is constructed by combining the current region and all regions in the previous frame that touch the current region. Object regions on a same trajectory are clustered as the same object. In the following, we formally describe the temporal clustering algorithm. We use the following definitions and notation.

Definitions and Notation

- R_i^t : Moving regions indexed by i at time t. Note that R_i^t must be spatially connected.
- $\mathbf{REG}^t$: Set of moving regions at time t.
- O_i^t : Moving object region indexed by i. $O_i^t = \{R_p^t,\ R_q^t\}$ indicates that the moving region O_i^t spread from an object. Since gaseous objects may spread to separated regions, O_i^t may be spatially disconnected.
- $\mathbf{OBJ}^t$: Set of moving objects at time t. If there exist m different moving objects, $\mathbf{OBJ}^t = \{O_1^t, O_2^t, \ldots, O_m^t\}$. $\mathbf{OBJ}^{(t-n)}$ is the memory of moving objects at n-frame behind from t frame time.
- OT_i^t : Region that O_i passed, that is, object's backward trajectory. If the object O_i's motion history from $t-n$ frame to t frame is $O_i^{(t-n)} \rightarrow O_i^{(t-n+1)} \rightarrow, \cdots \rightarrow O_i^t$, then $OT_i^t = O_i^t$, $OT_i^{(t-1)} = OT_i^t \cup O_i^{(t-1)}$, $OT_i^{(t-2)} = OT_i^{(t-1)} \cup O_i^{(t-2)}, \cdots$, $OT_i^{(t-n)} = OT_i^{(t-n+1)} \cup O_i^{(t-n)}$.
- $\mathbf{TRAJ}^t$: Set of backward trajectories at time t. If there are q different objects at time t, there exist q motion trajectories and so $\mathbf{TRAJ}^t = \{OT_1^t, OT_2^t, \ldots OT_q^t\}$.
- T_i^t : Index set of objects touching OT_i^t at time $(t-1)$. For example, if $O_p^{(t-1)} \cap OT_i^t \neq \emptyset$ and $O_q^{(t-1)} \cap OT_i^t \neq \emptyset$, then $T_i^t = \{p, q\}$. $T_i^t = \{p, q\}$ implies that $O_p^{(t-1)}, O_q^{(t-1)}$ are on O_i^t's motion path. So, $OT_i^{(t-1)}$ must be expanded to cover O_p^t, O_q^t, that is, $OT_i^{(t-1)} = O_p^{(t-1)} \cup O_q^{(t-1)} \cup OT_i^t$.
- B_i^t : Index set of trajectories connecting OT_i^t at frame time t. So, $B_i^t = \{k, l\}$ indicate that trajectories OT_i^t, OT_k^t, OT_l^t are branches from a source and so must be merged to a trajectory.

Based on these definitions and notation, we describe the temporal clustering algorithm as follows:

Moving Object Clustering Algorithm in the Temporal Domain

/* Initialization */

$\mathbf{TRAJ}^t = \mathbf{REG}^t$; /* So, $OT_i^t = R_i^t$. */

For time = t to t-1

  $\mathbf{NEWTRAJ} = \emptyset$;

  For i = 1 to $|\mathbf{TRAJ}^{time}|$ /* Check for each object trajectory */

    $T_i^{time} = \emptyset$;

    /* Expand a trajectory to cover objects touching the trajectory. */

/* Register objects touching the current trajectory. */

For p = 1 to $|\mathbf{OBJ}^{(time-1)}|$

If $\left(OT_i^{time} \cap O_p^{(time-1)} \neq \emptyset\right)$, $T_i^{time} = T_i^{time} \cup \{p\}$; End

End

/* Expand the current trajectory. */

$$OT_i^{time} = \left(\bigcup_{k=1}^{|T_i^{time}|} O_{T_i^{time}(k)}^{(time-1)}\right) \bigcup OT_i^{time};$$

/* Merge branch trajectories to the trajectory with the smallest index in the branches. */

/* Register indices of branch trajectories. */

For k = 1 to (i-1)

If $(OT_i^{time} \cap OT_l^{time} \neq \emptyset)$, $B_i^{time} = B_i^{time} \cup \{k\}$; End

End

/* Merge branches to a trajectory. */

$$OT_{B_i^{time}(1)}^{time} = \left(\bigcup_{k=1}^{|B_i^{time}|} OT_{B_i^{time}(k)}^{time}\right) \bigcup OT_i^{time};$$

$$\mathbf{NEWTRAJ} = \left(\mathbf{NEWTRAJ} \cup \{OT_{B_i^{time}(1)}^{time}\}\right) \setminus \left(\bigcup_{k=2}^{|B_i^{time}|} OT_{B_i^{time}(k)}^{time}\right);$$

End

/* Update object motion trajectories.*/

$\mathbf{TRAJ}^{(time-1)} = \mathbf{NEWTRAJ}$;

End

/* Trajectory Masking : Any moving regions enclosed in a trajectory must be clustered to a object. */

$\mathbf{OBJ^t} = \emptyset$;

For i = 1 to $|\mathbf{TRAJ}^{(t-2)}|$

$trac = \emptyset$; /* Index set of objects enclosed in the current trajectory */

For k =1 to $|\mathbf{REG}^t|$

If $\left(OT_i^{(t-2)} \supseteq O_k^t\right)$, $trac = trac \cup \{k\}$; End

End

$O_i^t = \bigcup_{k=1}^{|trac|} O_{trac(k)}^t$;

$\mathbf{OBJ^t} = \mathbf{OBJ^t} \cup \{O_i^t\}$;

End

/* Object Memory Update */

Spatial-Temporal Clusters at time t : $\mathbf{OBJ}^t$.

$\mathbf{OBJ}^{(t-2)} = \mathbf{OBJ}^{(t-1)}$;

$\mathbf{OBJ}^{(t-1)} = \mathbf{OBJ}^t$;

Figure 2 depicts the temporal clustering procedure. At each step, a trajectory is extended to include any objects touching the trajectory and any connected trajectories are merged to a single trajectory. At the trajectory masking procedure, the objects enclosed by a trajectory are clustered to the same object.

Figure 3 illustrates the result of a plume object segmentation. At first, motion estimation detects moving blocks. At the spatial clustering step, the moving blocks are clustered to moving regions. In figure 3(b), note how the moving regions are separated. Finally, the temporal clustering step clustered the moving regions to a single object.

III. Extracting Object Shape

As can be seen in Figure 3(c), the unit of resolution for segmented objects is a block. We need to extract actual object shapes from the block-based segmented regions. The first step for extracting object shape must be to detect object edges [22]. The edge detectors based on signal modeling using the maximum likelihood (ML) method [23] are not applicable to situations with complicated backgrounds. Therefore, we adopt a general gradient-based edge detector.

Unlike rigid objects or solid non-rigid objects, gaseous objects vary in density even within single-object regions and the edge intensity differs at different locations. So, an edge detector may detect only strong edges and detected edges may be an incomplete parts of an whole object's boundary. To construct a complete boundary, we further apply morphological filters [19].

A. Edge Detection

We use the gradient-based edge detector with the Sobel window. The Sobel windows for each direction are

$$\text{Horizontal } \mathbf{H}_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \text{Vertical } \mathbf{H}_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

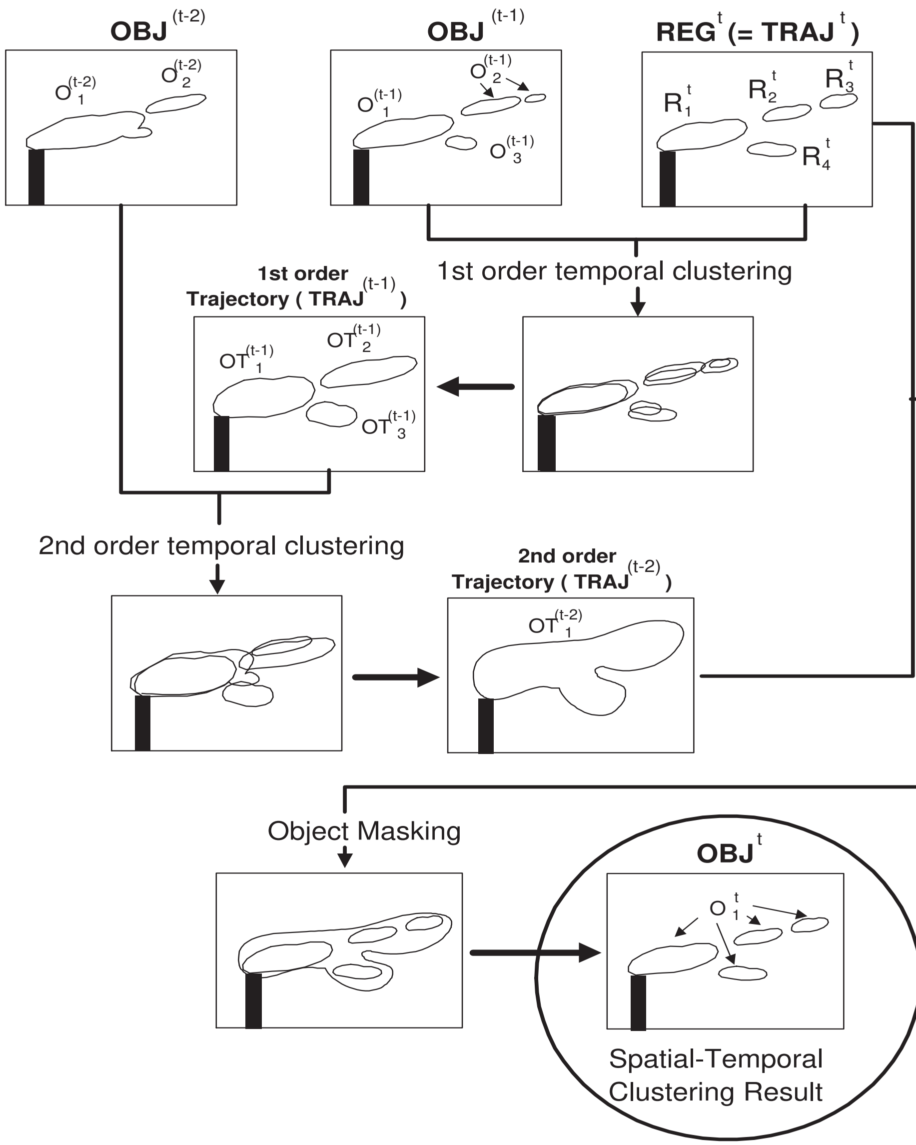


Fig. 2. Second order temporal Clustering Procedure

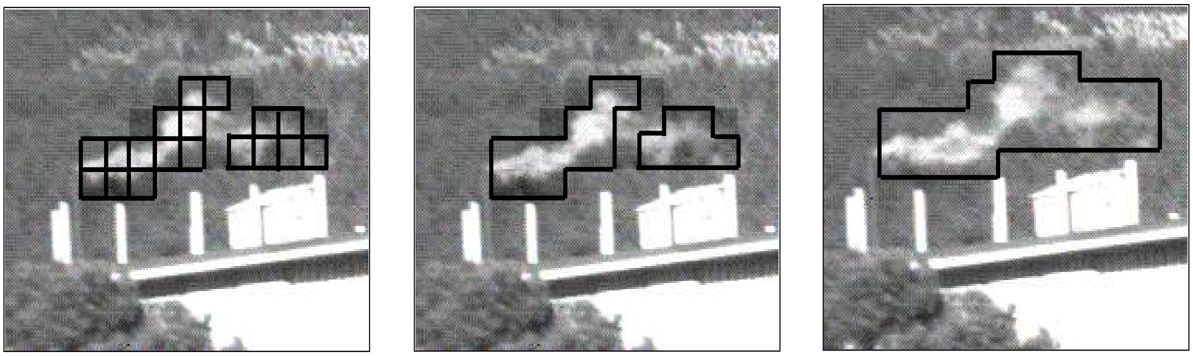
If the image intensity at (x, y) is $\mathbf{I}(x, y)$, the gradient operator at the location (x, y) is

$$\Delta_i(x, y) \equiv <\mathbf{I}, \mathbf{H}_i> (x, y) = \sum_m \sum_n \mathbf{H}_i(m, n) \cdot \mathbf{I}(x + m, y + n),$$

where Δ_1 and Δ_2 are the horizontal gradient and vertical gradient, respectively.
The gradient magnitude is then

$$|\mathbf{\Delta}(x, y)| = |\Delta_1(x, y)| + |\Delta_2(x, y)|.$$

For edge detection, we must set up a proper threshold for the gradient magnitude. If $|\mathbf{\Delta}(x, y)|$ exceeds a threshold, the pixel at (x, y) is declared as an edge pixel. Otherwise, the pixel is not



(a) Moving Blocks (b) Spatially Clustered Moving Regions (c) Spatially and Temporarily Clustered Object

Fig. 3. Result of a gaseous object segmentation.

an edge pixel. So, it is crucial to decide a proper threshold. If the threshold is too small, even noisy pixels may be declared as edge pixels. If the threshold is too big, many edge pixels are not detected. Figure 4 shows detected plume edges for various thresholds. Since different gaseous objects form different shapes and also have different intensities, we must adaptively decide the threshold in accordance with the object image's noise level and shape.

For a small threshold, noisy pixels are wrongly declared as edges so that such noisy pixels should increase the number of isolated edges. For a big threshold, only strong edges will be detected so that the number of edges should be small. The proper numbers of disconnected edges should be also proportional to object size, or the number of blocks covering the object. Based on these observations, the ratio between the number of disconnected edges and the number of blocks can be a criterion for deciding a proper threshold. When the ration is within a proper range, the threshold is also in a proper range.

In the following, we developed an algorithm that adaptively determines a proper edge threshold.

Algorithm for deciding edge-detection threshold

/* Define variables */

Object image : Object

Number of blocks covering object : NumOfBlk * NumOfBlk = |Object| *\

Number of disconnected edges : NumOfEdges

Edge-detector threshold : T_d

Edge detector for the object with threshold T_d: **Sobel**(Object,T_d)

Edges detected by **Sobel**(object, T_d) : Edges * NumOfEdges = |Edges| *\

Proper ratio factors : α, β $(\alpha < \beta)$

Edges = **Sobel**(Object, T_d) ;

NumOfEdges = |Edges| ;

While $\left(\left(\frac{\text{NumofEdges}}{\text{NumOfBlk}} < \alpha\right) \text{ or } \left(\frac{\text{NumofEdges}}{\text{NumOfBlk}} > \beta\right)\right)$

/* If the number of disconnected edges is too small compared to object size, dynamically decrease the threshold. */

If $\left(\frac{\text{NumofEdges}}{\text{NumOfBlk}} < \alpha\right)$

$$T_d = \left(1 - \frac{\alpha \cdot \text{NumofBlk} - \text{NumOfEdges}}{\alpha \cdot \text{NumofBlk}}\right) \cdot T_d;$$

End

/* If the number of disconnected edges is too large compared to object size, dynamically increase the threshold. */

If $\left(\frac{\text{NumofEdges}}{\text{NumOfBlk}} > \beta\right)$

$$T_d = \left(1 + \frac{\text{NumOfEdges} - \beta \cdot \text{NumofBlk}}{\beta \cdot \text{NumofBlk}}\right) \cdot T_d;$$

End

Edges = **Sobel**(Object, T_d) ;

NumOfEdges = |Edges| ;

End

A proper edge detection threshold : T_d

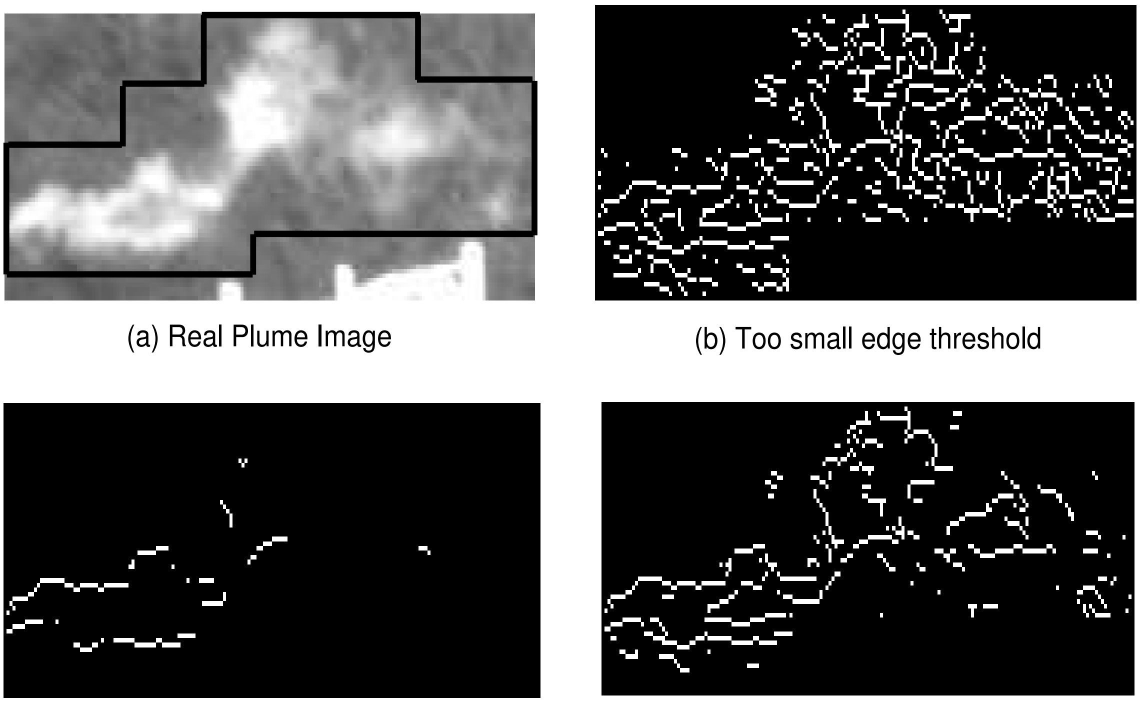
For our plume data, we set $\alpha = 2$ and $\beta = 3.5$. Figure 4(d) shows detected plume edges with the threshold obtained by the proposed algorithm.

B. Constructing Object Boundary and Region

Since an edge detector detects relatively strong edges, the detected edges for gaseous objects are usually disconnected. To construct the complete boundary from the disconnected edges, we apply morphological operations on the disconnected edges. We first remove very short edges because those edges are most likely noise. Then, we perform a dilating operation to connect nearby edges. Finally, we smooth the connected edges to produce a more natural boundary.

Let $\mathbf{X}$ be a binary image of detected edges. The pixel value '1' indicates a pixel on an edge. We define the following morphological operations [15]:

$$\text{clean}: \quad \mathbf{X} \oslash \mathbf{B}_p \quad \equiv \quad \{x : \ |(\mathbf{X} \setminus \mathbf{B}_c) \cap \mathbf{B}_c| > 3 \ \}$$



(a) Real Plume Image (b) Too small edge threshold

(c) Too large edge threshold (d) A proper edge threshold

Fig. 4. Results of edge detector with various thresholds.

$$\begin{aligned} \text{dilate}: \quad & \mathbf{X} \oplus \mathbf{B}_d \quad \equiv \quad \{x: \ \mathbf{X} \cap \mathbf{B}_d \neq \emptyset \ \} \\ \text{smooth}: \quad & \mathbf{X} \odot \mathbf{B}_b \quad \equiv \quad \{x: \ |\mathbf{X} \cap \mathbf{B}_d| \geq 3 \ \} \end{aligned}$$

where the windows are

$$\mathbf{B}_p = \begin{bmatrix} d & d & d \\ d & 0 & d \\ d & d & d \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B}_b = \begin{bmatrix} 1 & 0 & 1 \\ 0 & d & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

'd' means 'don't care' so that 'd' can be either 1 or 0. The operation pixel is the center of each window.

The morphological procedure for constructing object boundary from disconnected edges is

$$(((\mathbf{X} \oslash \mathbf{B}_c) \oplus \mathbf{B}_d)) \odot \mathbf{B}_b \ .$$

Figure 5 depicts each morphological step for constructing a plume boundary.

The next step is to recover the object region from the obtained boundary. If weak intensity regions are very long, even morphological operations could not recover the boundary. This indicates that the detected boundary may not completely enclose the interior of an object region.

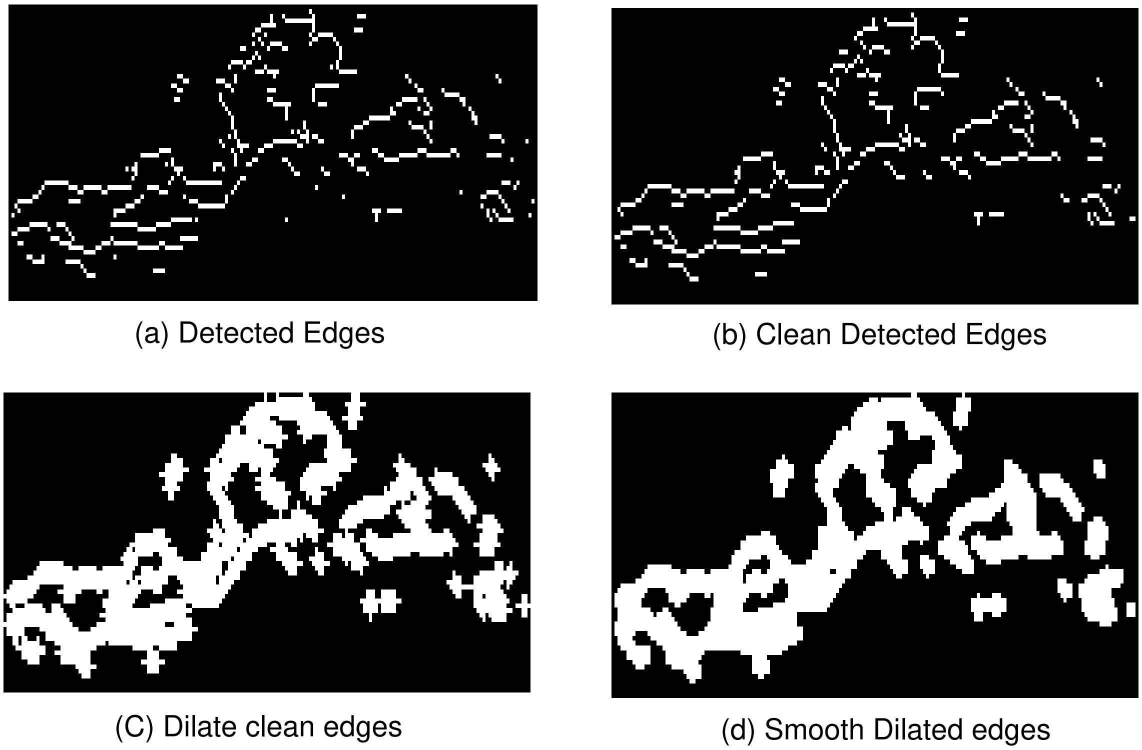


Fig. 5. Edge detection results for various thresholds.

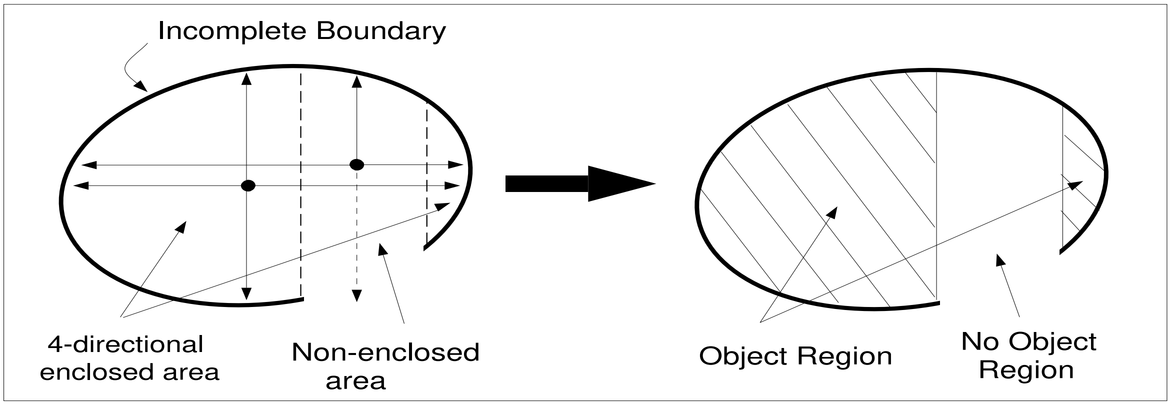
Thus, we must discriminate regions not enclosed by a boundary from separated regions. Our approach is to check enclosure at horizontal and vertical directions. Only if horizontal and vertical lines of a pixel meet the same boundary, is the pixel declared to be inside the object region. Figure 6 illustrates the proposed method to recover the object region from incomplete boundaries.

IV. Gaseous Object Identification

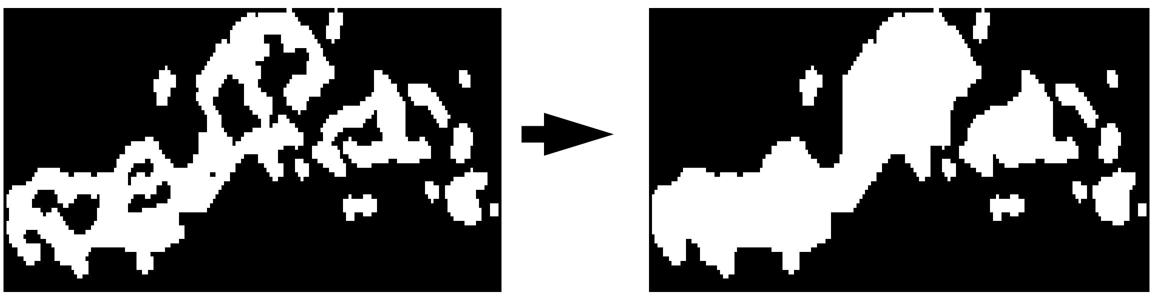
One feature that is commonly used for object identification is texture [20]. Since surveillance images like our plume images usually have low resolution and must be corrupted with noise, texture of those images are neither distinct nor discernible. Therefore, instead of texture we must exploit other features. The features we exploit are boundary direction pattern, frequency energy pattern and the object's eccentricity.

A. Boundary Pattern

Gaseous object shapes are arbitrary and irregular. The boundary directions of such irregular shapes must also be arbitrary. This implies that the directions between successive boundary pixels are much less correlated for gaseous objects than for rigid objects. We calculate the correlation of boundary directions to quantify the degree of shape irregularities.



(a) Object region reconsturction method



(b) Plume region reconstruction from incomplete boundaries

Fig. 6. Object region construction from incomplete boundaries

To record boundary directions, we use the 8-direction chain coding method. Figure 7 describes the chain code algorithm. Starting at a boundary pixel, the algorithm records the orientation of the next edge pixel. The boundary pixel orientations follow 8-neighbor connectivity. So, each direction can be labeled from 0 to 7 as shown in Figure 7. If a boundary has branches, there must be multiple directions at the boundary branches. In this case, the smallest label is chosen. The same process is continued until all boundary pixels have been visited.

Let $\theta(n)$ be a boundary direction code. Then the correlation coefficient at m-lag is

$$\phi(m) = \frac{\sum_{n=-\infty}^{+\infty} \theta(n+m)\cdot\theta(n)}{\sum_{n=-\infty}^{+\infty} \theta(n)^2}\ .$$

Figure 8 and 9 compares the correlation coefficients for gaseous objects and rigid objects. Table I compares one- and two lag correlation coefficients for objects in the figures. We measure the degree of the boundary irregularity with the average of one- and two lag correlation. That

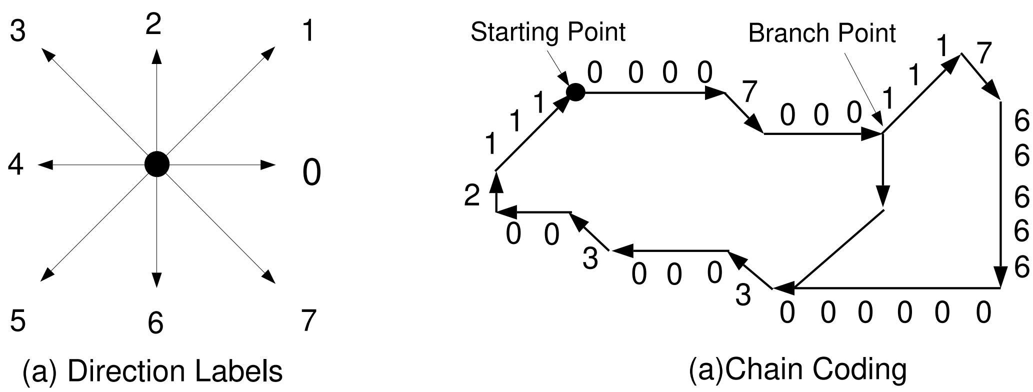


Fig. 7. Boundary chain coding. At branch point, the smallest labeled direction is chosen.

TABLE I

One- and two-lag correlation coefficients of boundary chain codes.

	Plume (Fig 8.(a))	Plume (Fig 8.(b))	House (Fig 9.(c))	Flag (Fig 9.(d))
One lag $\phi(1)$	0.66	0.64	0.9	0.78
Two lag $\phi(2)$	0.58	0.57	0.88	0.83

is, the boundary direction irregularity is

$$BC = \frac{\phi(1) + \phi(2)}{2} \quad .$$

B. Energy Pattern on frequency band

One characteristic of a gaseous object is that the densities are more non-uniform compared to rigid objects. This implies that gaseous objects are made of smooth regions in which the density gradually varies. So, a degree of smoothness can be a feature of gaseous objects. A legitimate way to determine an object smoothness is to investigate the object's energy distribution in the frequency domain. In smooth regions, energy in the middle frequency band must be relatively high compared to the low frequency band.

We use the wavelet transform to provide a spatial-frequency description of images [20]. We perform 3-level discrete wavelet decomposition with the Haar basis. Since target object sizes in surveillance imagery are small, the length of wavelet basis must be short enough to resolve small objects. Thus, the Haar basis must be one possible choice. Figure 10 depicts the 3-level wavelet decomposition and its subband labels. To remain the rotation-invariant, wavelet channels are combined to a scaling band and 3-band groups such as low- (channels 2-4), middle- (channels 5-7)

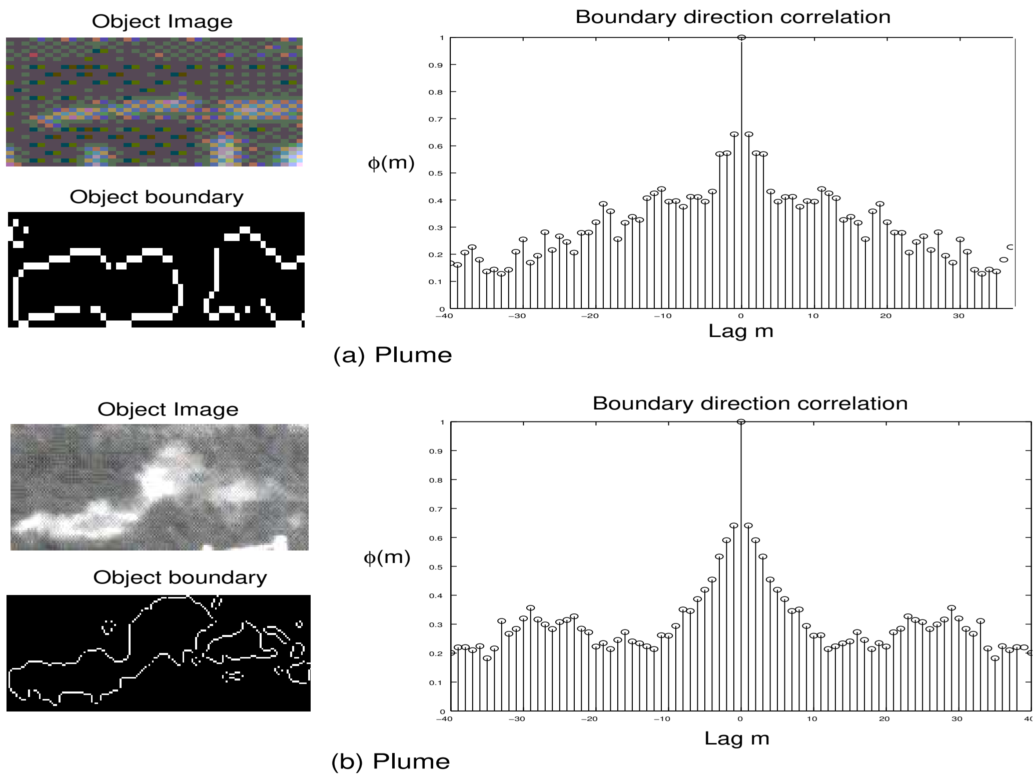


Fig. 8. Objects, their boundaries and the boundary chain-code correlations

and high-frequency bands (channels 8-10). The combined channel energy then reflects frequency energy distribution between smooth and rigid objects. So, the ratio of the scaling band energy plus the low channel energy to middle channel energy can be a criterion to distinguish smooth objects from rigid objects. Let $I_i(x, y)$ be the wavelet coefficient of subband i. The subband energy E_i is obtained as

$$E_i = \frac{1}{|R_i|} \sum_{(x,y)\in R_i} I_i(x,y)^2 \ ,$$

where R_i is an object region in subband i and is obtained in Section III.
Then, the subband energy ratio is

$$ER = \frac{E1 + E2 + E3 + E4}{E5 + E6 + E7}.$$

The smaller ER implies that the object is more likely to be a gaseous object.

Table II compares values of ER for various objects.

C. *Object Eccentricity*

Due to air flow, gaseous objects tend to spread to one direction. We measure the object spread with the best-fit ellipse eccentricity. The best-fit is the ellipse whose second moment is equal to that of the object. Figure 11 illustrates the best-fit ellipse for an object.

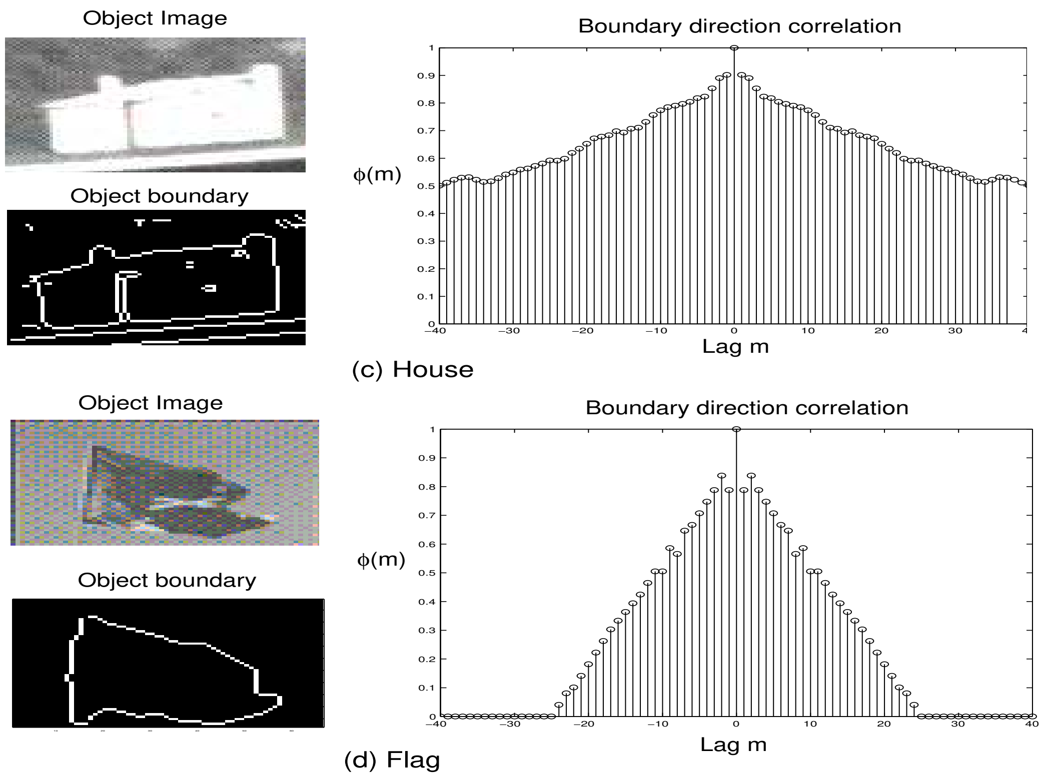


Fig. 9. (Continued) Objects, their boundaries and the boundary chain-code correlations

TABLE II

WAVELET-BASED SUBBAND ENERGY RATIO

	Plume (Fig 8.(a))	Plume (Fig 8.(b))	House (Fig 9.(c))	Flag (Fig 9.(d))
Energy Ratio(ER)	11.5	13.2	108	80.6

For an object region R containing $|R|$ pixels, we obtain the eccentricity through sequential calculations as following;

1. Central moments :

$$\text{Center of x mass : } \overline{X} = \frac{1}{|R|} \sum_{(x,y)} \sum_{\in R} x, \quad \text{Center of y mass : } \overline{Y} = \frac{1}{|R|} \sum_{(x,y)} \sum_{\in R} y.$$

The (m, n) order central moments :

$$\mu_{m,n} = \frac{1}{|R|} \sum_{(x,y)} \sum_{\in R} (x - \overline{X})^m (y - \overline{Y})^n \ .$$

2. Orientation : Orientation is the angle to the axis with the smallest inertial moment. The

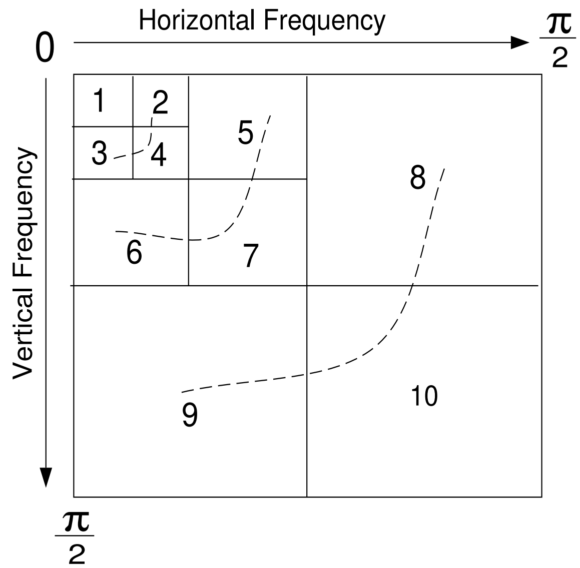


Fig. 10. Wavelet decomposition and its subband labels and channel combinations.

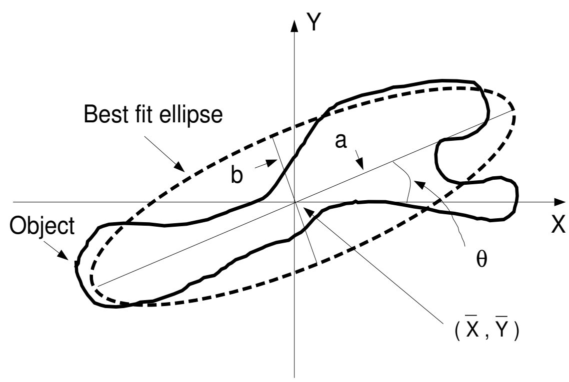


Fig. 11. Best fit ellipse.

orientation is obtained as

$$\theta^* = \arg\min_\theta I(\theta) = \arg\min_\theta \left\{ \sum_{(x,y)} \sum_{\in R} [(y-\overline{y})\cos\theta - (x-\overline{x})\sin\theta]^2 \right\}$$

The solution is

$$\theta^* = \frac{1}{2}\arctan\left[\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right] .$$

3. Best-fit ellipse : Let a and b denote the lengths of the major and the minor axes, respectively. The least and the greatest inertial moments of an ellipse are

$$I_{min} = \frac{\pi}{4}ab^3, \qquad I_{max}\frac{\pi}{4}a^3b$$

which gives

$$a = \left(\frac{\pi}{4}\right)^{1/4}\left[\frac{I_{max}^3}{I_{min}}\right]^{3/8}, \qquad b = \left(\frac{\pi}{4}\right)^{1/4}\left[\frac{I_{min}^3}{I_{max}}\right]^{3/8}$$

For orientation θ^* ,

$$I_{min}(\theta^*) = \sum_{(x,y)} \sum_{\in R} [(y-\overline{y})\cos\theta^* - (x-\overline{x})\sin\theta^*]^2$$

$$I_{max}(\theta^*) = \sum_{(x,y)} \sum_{\in R} [(y-\overline{y})\sin\theta^* + (x-\overline{x})\cos\theta^*]^2$$

4. Eccentricity : $EC = \dfrac{b}{a}$.

Small EC indicates long spread of an object. So, an object with smaller EC has a greater chance of being a gaseous object.

V. Test Results

We test the proposed system for detection of plumes. We use a hierarchical classifier for plume identification. For the plume classifier, we set the feature criteria as

$$\begin{aligned} \text{Boundary direction correlation :} \quad & BC \leq 0.75 \\ \text{Subband energy ratio :} \quad & ER < 30 \\ \text{Object eccentricity :} \quad & EC < 1/3 \end{aligned}$$

If all BC, ER and EC satisfy the conditions, the object is declared a plume. If BC satisfies its condition and either ER or EC satisfies the condition, the object is suspected as a possible plume. Therefore, BC is used as the main feature. Figure 12 shows plume detection results. The macro block size is 16×16. The purple objects indicate plumes, and the blue ones are plume suspects. Like the plumes in Figure 12(a),(b), the plumes spreading more in one direction due to strong winds are easily detected. This is because strong wind makes the plume movement apparent. Where there is no wind flow or very little air density variation, plumes move very slowly. Then it is difficult to detect such plumes. However, even in strong wind, the plume in Figure 12(c) does not satisfy the eccentricity condition. The plume volume is too small compared to wind speed and so the plume quickly disappears. In Figure 12(c), it must also be noted that the proposed spatial-temporal segmentation method dose a good job clustering the separate plume regions into a single plume. The plume in Figure 12(d) is behind the chimney; this plume does not satisfy the eccentricity condition.

In Figure 12(a), a small plume on the right side is missed. The 16×16 macro block is too big to detect the small plume's motion. To increase the motion estimation resolution, we reduce the block size to 12×12. Then the small plume was also detected as shown in Figure 13. The small plume was detected as a suspected one and encircled in the figure. For the 12×12 blocks, the block noise threshold is still 90 % confidence, same as for 16×16 blocks. This means that the

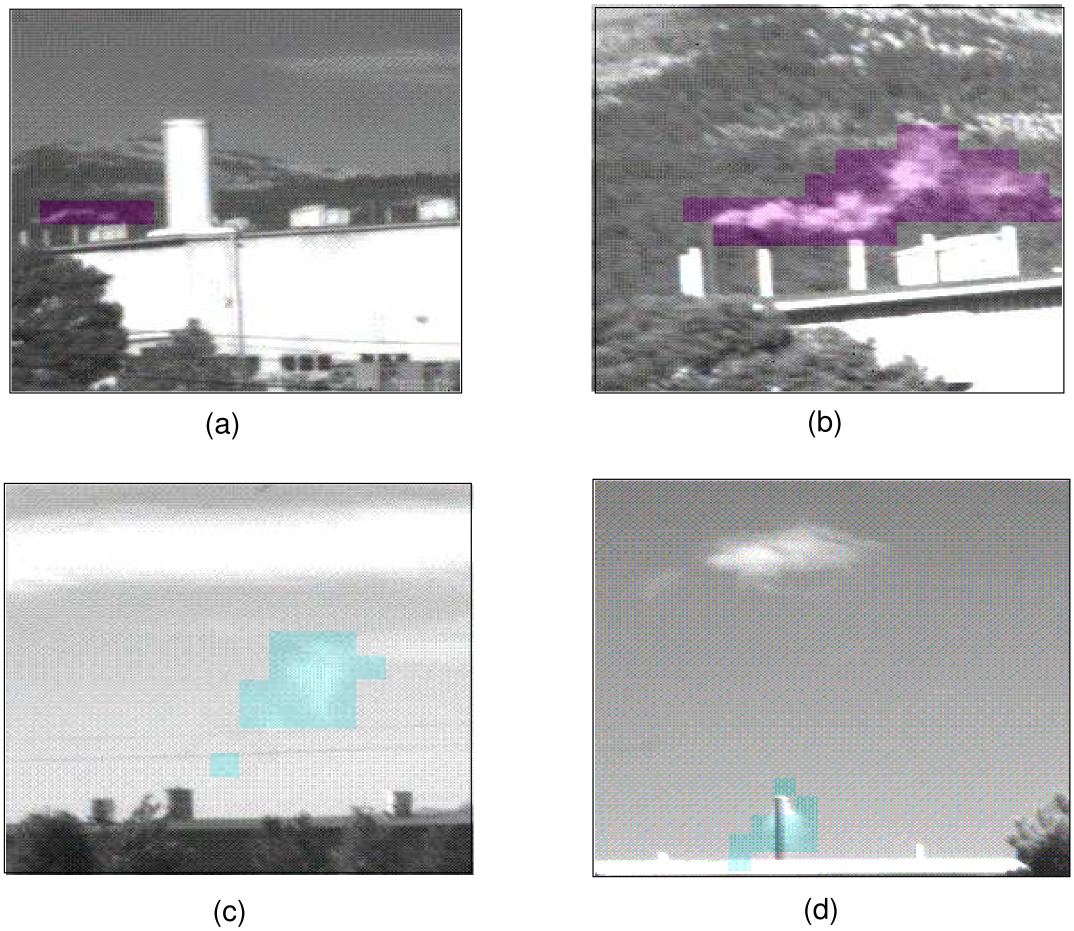


Fig. 12. Plume detection results. The macro block size is 16x16.

threshold for 12×12 blocks is $1.8 \cdot \sigma$ where the threshold for 16×16 blocks is $1 \cdot \sigma$. However, if we further reduce the block size of 6×6, the proposed method fails to detect any plumes because of too many noise blocks. This implies that the confidence test for noise threshold does not work if the blocks are too small blocks. So, we need to find the proper block size which can detect small motion object but is not vulnerable to noise. The issue of standard versus non-standard block size is also a relevant engineering design consideration.

VI. Conclusion

In this work, we develop a system for tracking a gaseous object. The proposed system well track gaseous objects for which while conventional methods such optical flow method or model-based method often fail to track those objects. It is also more robust to noise and faster than the conventional optical flow method. In addition, the proposed method embedded for operational systems can also be implemented via an MPEG-4 code-stream.

Fig. 13. Plume detection with 12x12 macro block.

The developed system was applied to detect smoke plumes. The system performance was affected by wind strength, plume size, plume volume and motion estimation block size. However, the proposed system can detect most of the plumes that humans can detect.

In this work, we assume that the camera platform is fixed and ground based. In real applications, the camera may be moving and jittering. Therefore, we must also develop a system to compensate the camera motion and the jitter noise.

References

[1] MPEG video group, *MPEG-4 Overview*, ISO/IEC JTC1/SC29/WG11 N2725, Mar. 1999.

[2] S.C Han, *Object-Based Representation and Compression of Image Sequences*, Ph.D thesis, Rensselaer Polytechnic Institue, Troy, New York, Sept. 1997.

[3] MPEG video group, *Information technology - Generic coding of audio-visual objects- Part2:Visual*, ISO/IEC JTC1/SC29/WG11 N2688, Mar. 1999.

[4] C. Kervrann and F. Heitz, "Statistical deformable model-based segmentation of image motion", *IEEE Trans. on Image Processing*, vol. 8, no. 4, pp. 583–588, Apr. 1999.

[5] P. Anandan, J.R. Bergen, and K.J. Hanna, *Hierachical Model-Based Motion Estimation in Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, 1993.

[6] A.M. Tekalp, *Digital Video Processing*, Prentice Hall, 1995.

[7] T. Tsai and T. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch", *IEEE Trans. on Acoust., Speech and Siganl Processing*, vol. 29, pp. 1147–1152, Dec. 1981.

[8] J.L. Mitchell, W.B. Pennebaker, C.E. Fogg, and D.J. Legall, *MPEG video compression standard*, Chapman Hall, 1996.

[9] B.G. Haskell, A.P. Puri, and A.N. Netravali, *Digital Video: An introduction to MPEG-2 video compression standard*, Chapman Hall, 1997.

[10] B.K.P Horn and B. Schunk, "Determining optical flow", *Artificial Intelligence*, vol. 17, pp. 309–354, 1981.

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[12] V. Dang, A. Mansouri, and J. Konrad, "Motion estimation for region-based video coding", *IEEE ICIP*, pp. 189–192, 1995.

[13] C. Stiller, "Objecte-based estimatation of dense motion fields", *IEEE Trans. on Image Processing*, vol. 6, no. 2, pp. 234–250, Feb. 1997.

[14] M. Kardouchi, A. Diapanda, and L. Legrand, "Motion estimation with Markov random fields by using a direct algorithm", *Proceedings of the SPIE*, vol. 3101, pp. 95–103, 1997.

[15] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[16] G. Adiv, "Determining three-dimensional motion and sturcture form optical flow generated by serveral moving objects", *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 7, pp. 384–401, July 1985.

[17] J. Ostermann, "Object-based analysis-synthesis coding (obasc) based on the sourcer model of moving flexible 3-d objects", *IEEE Trans. on Image Processing*, vol. 3, pp. 705–711, Sept. 1994.

[18] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields", *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 14, pp. 910–927, Sept. 1992.

[19] P. Salembier and M. Pards, "Hierachical morphological segmentation for image sequence coding", *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 639–651, Sept. 1994.

[20] M. Unser, "Texture classification and segmentation using wavelet frames", *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.

[21] T. Aach and A. Kaup, "Statistical model-based change detection in moving video", *Signal Process(Elsevier)*, vol. 31, pp. 165–180, Mar. 1993.

[22] L.S. Davis, "A survey of edge detection techniques", *Computer Graphics and Image Processing*, vol. 4, pp. 248–270, 1975.

[23] T.N. Pappas and J.S. Lim, "A new method for estimation of coronary artery demensions in angiogram", *IEEE Trans. on Acoust., Speech and Siganl Processing*, vol. 36, pp. 1501–1513, Sept. 1988.