Region-Based Approaches to Visual Motion Correspondence

Chiou-Shann Fuh, Petros Maragos, and Luc Vincent

Harvard Robotics Laboratory

Technical Report 91–18

November 1991

Region-Based Approaches to Visual Motion Correspondence

Chiou-Shann Fuh, Petros Maragos, and Luc Vincent

Harvard Robotics Laboratory
Division of Applied Sciences
Pierce Hall
Cambridge NA 02138

Abstract

This paper presents a correspondence method to determining motion displacement fields in sequences of intensity images where the motion tokens to be matched between consecutive image frames are 2-D regions. These regions contain perceptually important image features. The computation of the 2-D image velocity field is done in three stages: region extraction, region matching, and velocity smoothing. The emphasis of the work is on the region extraction part, where four possible approaches are developed and compared. Thus, in each image frame the regions are extracted either from the sign representation of the image convolution with a Laplacian of a Gaussian, or by thresholding the output of morphological image transformations for peak/valley detection, or by performing morphological graylevel watershed segmentation, possibly followed by further segmentation of the resulting binary regions via the watershed of their distance transforms. For region matching, a general correspondence approach is applied to the region tokens by using similarity criteria based on region features. Image velocities are then identified as the spatial vector displacements among centroids of corresponding regions. The computation is completed by smoothing the initial velocity field with a spatio-temporal vector median filter. The performance of these four approaches for region extraction and matching is evaluated in the presence of noise. Overall, the proposed region-based methods for computing image velocities are simple, efficient, less computationally complex than intensity correlation methods, and (as our experiments on real images indicate) more robust than iterative gradient methods especially for medium or long-range motion. In addition, the developed morphological region extraction methods provide several robust 2-D image features to be used in visual correspondence problems.

Index terms: computer vision, motion analysis, correspondence, mathematical morphology.

This work was supported by the National Science Foundation under Grant MIPS–86–58150 with matching funds from DEC and Xerox, and in part by the Army Research Office under Grant DAALO3-86-K-0171 to the Brown-Harvard-MIT Center for Intelligent Control Systems.

1 Introduction

Motion analysis is a major task of computer vision systems that attempt to extract information from moving imagery. It deals with general problems such as the detection and measurement of 2-D image motion and/or the recovery of the 3-D motion and shape of object surfaces given a spatio-temporal signal $I(x, y, t)$ of 2-D intensity images. There has been voluminous amount of work on visual motion, and some reviews or detailed discussions on this topic include [2, 10, 11, 13, 23]. When objects are being imaged through a camera (or a human retina) moving relative to the objects, the apparent motion of brightness patterns corresponds to a 2-D image velocity field (often called 'optical flow'). This is represented by a 2-D spatio-temporal vector field (v_x, v_y), where v_x, v_y denote velocities in x, y direction. For discrete-time sequences of image frames an average value (over the interframe time sampling period Δt) of this velocity field is provided by the displacement vector field (divided by Δt) whose vectors determine the correspondences among image points of successive image frames. In some cases the image velocity field may not generally be equal to the true 2-D motion field that results from projecting the 3-D motion onto the image plane. Nevertheless, due to its accessibility and the rich information it contains [11, 16, 19, 23, 41, 48], the measurement of the image velocity field is very important in motion analysis. For instance, there are many approaches to 3-D motion and shape recovery which assume that 2-D velocity data (sparse or dense) have been obtained in advance; examples include [15, 47, 48, 49]. 2-D velocities can also be used to track the motion of objects projected onto the image plane. In addition, measuring the displacement field is necessary to provide the necessary motion compensation for the interframe prediction and interpolation tasks encountered in video data compression; examples can be found in [14, 29] and the review [28].

The major approaches to computing 2-D image velocity fields can be roughly classified as either using *gradient* models or *correspondence* of motion tokens. Most gradient models are based on some constraints or relationships among the image spatial and temporal derivatives. Examples include the optical flow constraint $dI/dt = 0 \Longleftrightarrow \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y = -\frac{\partial I}{\partial t}$ proposed in [12], a least-squares approximation of optical flow by affine vector fields using shape gramians developed in [6], and various smoothness constraints used to derive optical flow along contours [9, 50]. Another broad class of gradient-based methods are all the pixel-recursive algorithms as in [29], popular among video coding researchers. Although gradient models are analytically more tractable, may lead to iterative local image operations, and can provide spatially dense velocity estimates, they are computationally intensive, apply only to short-range motion, and are highly susceptible to noise due to the frequent usage of derivatives and their discrete approximations. By contrast, the correspondence methods are more immune to noise and can be also applied to both short- and long-range motion. They are based on matching and tracking over time simple tokens (sets of elementary image features) in one image frame with their counterparts on the same object in subsequent frames. These tokens need not have more than a 2-D structure as argued in [41] and usually are any of the following three kinds: (i) isolated *points*, e.g., corners, curvature inflection points, or other interesting points representing important image features [4, 40, 41]; (ii) short *curves*, e.g., edges or other line segments [9, 23, 41, 50]; (iii) blob-like *regions* representing small connected parts of the image with similar brightness or texture [7]. The main difficulty of all correspondence methods lies in reliably extracting good features and tracking them. After the correspondence has been solved, the resulting displacement vectors between corresponding tokens serve as (scaled) sparse estimates of the velocity at these tokens.

This paper presents a correspondence approach to measuring the 2-D image velocity field by using *regions* as the simple tokens to extract from each image frame and track over time. In recent research considerable attention has been given to edges (e.g., zero-crossings of the Laplacian of a Gaussian [24]) as being perhaps the most desirable features to match in binocular stereopsis or motion analysis [23]. However, without doubting the general usefulness of edges as important image

features, we view the region matching as more robust than edge matching, because noise perturbs the coherence of a region less than its boundaries (edges). This was demonstrated by Nishihara [30] who solved the correspondence problem for binocular stereo by cross-correlating the binary regions (sign areas) bounded from the zero-crossing contours of the band-pass filtered images $\nabla^2 G * I$. (∇^2 is the operator $\partial^2/\partial x^2 + \partial^2/\partial y^2$, $G(x,y) = \exp[-(x^2+y^2)/2\sigma^2]/2\pi\sigma^2$ is a Gaussian function with standard deviation (scale parameter) σ, and $*$ denotes 2-D convolution.) Mayhew and Frisby [25] have also found that intensity edges cannot by themselves disambiguate some correspondences in binocular stereopsis unless they are supplemented by region features such as intensity peaks and valleys. Additional strong evidence for the possible effectiveness of blob-like regions is provided by the psychophysical experiments of Ramachandran and Anstis [31], which demonstrated that the human visual system during its first short-term phase of perceiving apparent motion is more likely to detect correspondences between regions of similar brightness or texture before it detects sharp outlines or edges.

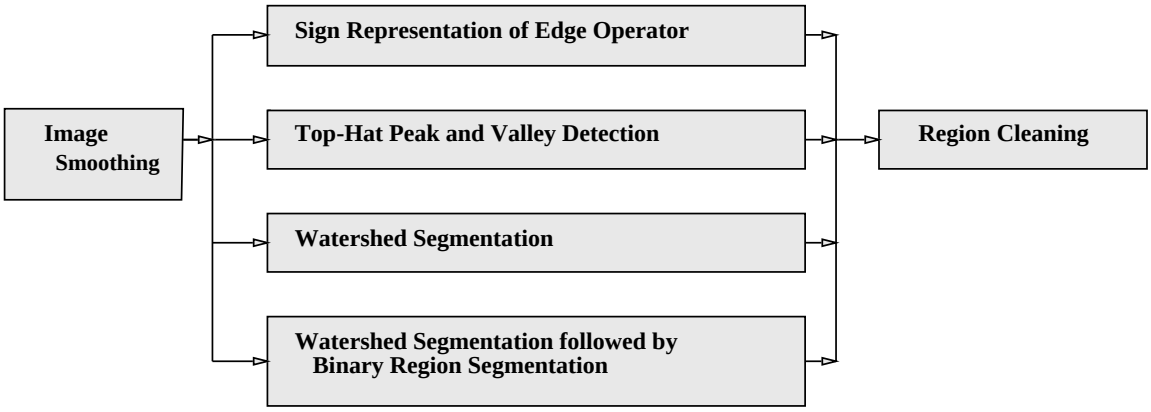
Motivated by all the above evidences, in this paper we study and compare several region-based approaches to motion correspondence. As Figure 1 summarizes, the common procedure in all our approaches consists of three stages: (I) *Region Extraction:* This part (discussed in Section 2), which carries the main emphasis of our paper, deals with pre-cleaning the image, extracting the regions, and cleaning these regions. We study four different approaches to extracting regions: sign representation of the image convolution with $\nabla^2 G$, morphological peak/valley detectors, morphological image segmentation by watersheds, and watershed segmentation of distance functions of binarized regions resulting from segmentation of graylevel images. Although the sign of the convolution with $\nabla^2 G$ offers reasonably effective regions, the operators and algorithms of mathematical morphology for feature extraction and segmentation [22, 27, 35, 43] have the advantage of providing multiscale region features without blurring their boundaries. Thus, our preferance for using morphological feature extraction and segmentation approaches to extract regions is based on the inherent ability of morphological operators to easily relate to shape and hence to provide regions that may correspond to more easily identifiable subparts of the moving object. II) *Region Matching* (discussed in Section 3), where Ullman's general correspondence theory [41] is applied to region tokens by using several similarity criteria for matching. These criteria are based on a more extended set of region features than the affinity measure used in [41]. After the region matching, velocity estimates are then identified as the spatial displacements among centroids of corresponding regions. III) *Velocity Smoothing* (discussed in Section 3), where the 2-D velocity data are smoothed with a spatio-temporal vector median filter.

Finally, Section 4 discusses the main conclusions reached via some numerical experiments we conducted to compare the performance of the four approaches under salt-and-pepper and white Gaussian noise.

2 Region Extraction Algorithms

In this section, we are concerned with the extraction of the regions which are then to be matched over time in the image sequence. We define as *regions* connected sets of pixels (x,y) which are either subsets of the spatial image domain with similar brightness or contrast, or are bounded by some contour of maximum intensity gradient points, or belong to the same intensity peak/valley. The present task can be viewed as (and in the watershed approaches it is actually) an image segmentation. Our present goal is not necessarily to obtain a very precise segmentation of the image under study: the only important considerations are that the obtained regions should approximately convey the aforementioned types of information, be easily and accurately matchable from one image to the next one, and be extractable with efficient algorithms. Thus, the segmentation process needs not necessarily be very precise. The scale (region size) at which the segmentation occurs is also

(a) Major steps of region-based visual motion correspondence



(b) Details of region extraction

Figure 1 : Motion analysis system

an important consideration with conflicting issues: Small scale regions provide denser velocity estimates, but their matching may be more difficult (due to the large number of possible matches) and hence less reliable. Larger scale regions provide more robust features and velocity estimates, but lead to sparser velocity fields. Of course, an "optimum" region size will depend on the particular application and type of moving imagery. Our algorithms provide the ability to control the region size through either the smoothing filters related to edge/peak/valley detection or the marker features needed by the watershed segmentation. In this section, four different region extraction techniques are considered:

- sign representation of convolution with Laplacian of Gaussian
- morphological peak/valley extraction via top-hat transformation
- watershed segmentation with markers
- watershed segmentation followed by binary region segmentation

The above approaches should also be as nonspecific as possible: we do not want to develop highly sophisticated procedures which will only provide good results for a particular image type. On the contrary, our motion estimation problem has to be considered in its full generality. This is the reason why we purposely do not take into account any characteristics specific to our set of test images.

The above segmentation approaches are described in sections 2.2 to 2.5. Prior to their execution, a first step consisting of an image smoothing is used, which is described in section 2.1. Additionally, the results provided by each of the algorithms described below need to be cleaned before applying the matching procedures. This region cleaning step is explained in section 2.6.

2.1 Image Smoothing

To avoid spurious regions due to small-scale noise that may affect the region extraction algorithms, we pre-smooth each image frame by using a morphological filter belonging to the category of the so-called *Alternating Sequential Filters (ASFs)*. These operators are based on iterations of alternating openings and closings by structuring elements of increasing size. However, ASFs tend to preserve

the edges better than median-type filters, yield a fixed point (i.e., a smoothed image invariant to further applications of the filter) in a single pass, and provide much more flexibility and control. They have been initially described and used in [38]. A single-scale version of ASFs was analyzed in [21] and was closely related to median filters. Their general theoretical properties have been formally studied in [36, ch.10]. Among others, the good behavior of these filters in the presence of noise is now well-known [33, 34, 37, 39]. In fact, they are somewhat the "general-purpose" smoothing filters of mathematical morphology; i.e., when the characteristics of the noise are unknown, or when the segmentation problem is not clearly specified, the ASFs is a very good choice to try first because they can provide sequential smoothing at multiple scales.

For these reasons, ASFs seem to be particularly suited to our present problem. Here, to be even more general, we chose as building blocks for these filters a maximum of openings by line segments of different orientations, and the dual closing. This allows us to better preserve the line-type image structures, which usually convey important information. More specifically, let us denote by S_i, $i = 1, .., 4$, the line segments shown in Figure 2. If $I(x, y)$ is a single-frame image signal, let $I \ominus B$, $I \oplus B$, $I \circ B = (I \ominus B) \oplus B$, and $I \bullet B = (I \oplus B) \ominus B$ denote the morphological erosion, dilation, opening, and closing of I by a structuring element B (a set of pixels); for properties and applications of these operations see [21, 22, 35, 37]. If each S_i is viewed as a unit-size element, then

$$nS_i = \underbrace{S_i \oplus S_i \oplus \cdots \oplus S_i}_{n \text{ times}}$$

denotes its corresponding element of size $n = 1, 2, 3, ...$ The openings γ_n and closings ϕ_n that make up the filter are the following:

$$\gamma_n(I)(x, y) = \max_{i \in [1,4]} \{I \circ nS_i(x, y)\} \quad (1)$$

$$\phi_n(I)(x, y) = \min_{i \in [1,4]} \{I \bullet nS_i(x, y)\} \quad (2)$$

Figure 2: *Structuring elements S_i for image smoothing.*

In the actual filter, we have to start either with an opening, or with a closing. According to our tests, this does not change much the resulting image in most cases, so we arbitrarily decided to start with a opening. The filter ψ we ended up using is therefore given by the following cascade (i.e., operator composition)

$$\psi = \phi_n \gamma_n \ldots \phi_2 \gamma_2 \phi_1 \gamma_1 \quad (3)$$

The maximum size n of the filter ψ is an important parameter. In some cases, the critical size can be determined either from the noise statistics [33], or from morphological size distributions [34], or through the quality of the resulting segmentation. Here, we do not address such problems and fix the maximum size to 3.

2.2 Sign Representation of $\nabla^2 G * I$

Regions are the complementary representation of edges. Hence, they can be obtained from edge operators. Specifically, the edge detection operation [24] $\nabla^2 G * I$ is applied to each image frame I. Binary edge information is obtained by the zero-crossing contours of the operator's output. For

each image frame, the regions are identified as the connected subsets of the image plane whose boundaries are these edge contours. Thus, the set of image pixels at which this edge signal has a positive sign identifies the collection of *positive regions*, and its set complement yields the *negative regions*. There is a trade-off in selecting a value for the scale parameter σ. For large σ, the regions are large, and their number per frame is small. To achieve dense velocity estimates, small values of σ are preferred. On the other hand, to achieve a matching that is more robust and less susceptible to noise, a larger σ is preferred. In our experiments we implemented the $\nabla^2 G$ as the difference of two Gaussians, one (the excitatory) with $\sigma = 2.25$ and another (the inhibitory) with $\sigma = 0.75$; the size of the convolution kernel was 9×9 pixels. Examples of extracted regions are shown later. The region extraction process is completed by *labeling connected components*, where at each time t_k, each positive (or negative) region has been identified as a connected component of the binary image representing the positive (or negative) sign of $\nabla^2 G * I(x, y, t_k)$.

Among various alternative edge operators whose sign representation can provide region features, the $\nabla^2 G * I$ approach is chosen mainly because it gives closed edge curves. However, there are also several morphology-based edge operators with the same property [5, 18, 35, 42]. For example, in [8] we have also experimented with regions extracted as the sign representation of edges obtained via the morphological "Laplacian"-like edge operator of [42]. The matching results based on these nonlinear edge operators were very similar with those obtained using the linear $\nabla^2 G$ operator.

2.3 Binarized Peak/Valley Detection Transformations

If I is the intensity image at some time frame, two morphological operators that can extract its intensity peaks and valleys, respectively, are the opening and closing residuals [26, 35] (known as "top-hat" transformations and due to Meyer):

$$Peak(I) = I - (I \circ B) \geq 0 \quad (4)$$

$$Valley(I) = (I \bullet B) - I \geq 0 \quad (5)$$

where B is a flat convex structuring element. The opening $I \circ B$ smooths I by cutting down its peaks; hence the residual signal $Peak(I)$ contains only the peaks of I. The shape and size of B control the shape and maximum size of of the binary regions of support of these peaks. Similarly for the valleys. In our experiments, we use as structuring element an octagon $S = \{(x, y) : x^2 + y^2 \leq 5\}$ of size 2; i.e., $B = S \oplus S$. Note that the resulting element B has the same size as the truncated impulse response for the $\nabla^2 G * I$ operation used in Section 2.2, so that both the linear smoothing via the Gaussian convolution and the morphological smoothing via the opening or closing refer to the same scale.

The value of $Peak(I)$ at a certain pixel location determines the contrast (or the "strength") of the peak at that location. We produce binary *peak regions* by thresholding at level T, i.e., by setting all pixels (x, y) at which $[\text{Peak}(I)](x, y) \geq T$ equal to 1 and 0 elsewhere. It is not a simple task to find an optimum T for general-purpose detection. In the approach we used, all the nonzero values of the peak signal $Peak(I)$ were sorted for each frame and T was selected as the 70% percentile value. (The value 70 was experimentally found to give reasonable results.) Similarly, the binary *valley regions* result from thresholding the valley signal Valley(I) at T. Figures with examples from the above peak/valley region extraction will be shown later.

2.4 Watershed Segmentation

The techniques described in this section and the following one will be illustrated on Figure 4.a. Here, we shall make use of one of the most powerful tools provided by mathematical morphology, namely the watershed transformation [5]. It is defined for grayscale images via the notion of a *catchment basin:* let us regard the image under study as a topographic relief and assume it is

raining on it. A drop of water falling at a point p flows down along a steepest slope path until it is trapped in a minimum m of the relief. The set $C(m)$ of the pixels such that a drop falling on them eventually reaches m is called catchment basin associated with the minimum m. The set of the boundaries of the different catchment basins of an image constitute its *watersheds*. These notions are explained in extensive details in [27, 46].

In other words, the watershed edges or lines are located on the crest-lines of the image which actually separate two different minima (the watershed elements are always closed edges). The basic idea of watershed segmentation consists therefore in applying this tool to the gradient of the image I to be segmented. This is illustrated by Figure 3. Note that by *gradient,* we mean here a morphological gradient of I, i.e., an image where the gray-level of each pixel is indicative of the slope in the original image. One of the most popular gradients, often referred to in literature as Beucher's gradient [35], is obtained by taking the algebraic difference between an elementary dilation and an elementary erosion of I:

$$grad_B(I) = (I \oplus B) - (I \ominus B), \tag{6}$$

(with B being an elementary square or disc).

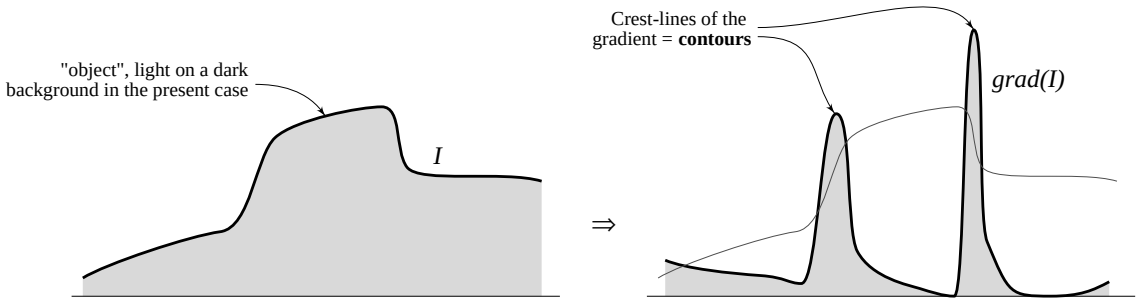


Figure 3: *Grayscale segmentation by watersheds of the gradient.*

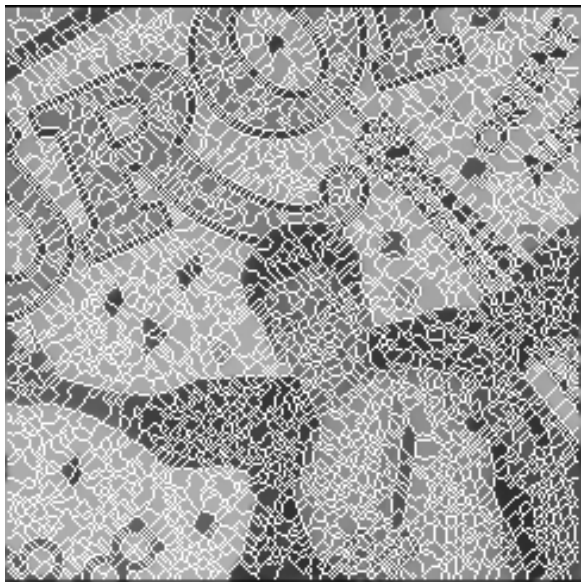
Now, as explained in [27, 45, 46], the direct application of the watershed transformation to a gradient image usually leads to poor results. Indeed, even after dramatic filtering of the original image or of its gradient, the latter often exhibits far too many minima, and thus far too many catchment basins. Hence, straightforward watershed segmentation of the gradient mostly leads to oversegmented images, as illustrated by Figure 4.

To get rid of this problem, one of the best solutions consists in making use of markers of the regions to be extracted. By marker of a given region, we mean a connected component of pixels located inside this region. The assumption used here is that it is easier to design robust methods to extract markers than to directly extract the precise contours of the desired regions. Once these markers have been extracted, a classical morphological procedure based on grayscale geodesic erosions [17, 45] allows us to:

- impose these markers as minima of the gradient while removing all other minima by filling up their associated catchment basins,
- preserve the highest crest-lines of the gradient located between two markers.

Then, computing the watersheds of the modified gradient provides the best possible contours with respect to the set of input markers and to the gradient itself. The quality of the resulting segmentation is directly related to the initial markers and to a certain extent, to the used gradient. This methodology, which is detailed in [46], has already proved to be particularly useful in various fields of image analysis, ranging from medical imaging to material sciences, remote sensing and digital elevation models.

(a)



(b) *(c)*

Figure 4 : (a) Original image. (b) Morphological gradient of (a). (c) Image oversegmentation resulting from computing the watersheds of the gradient (b).

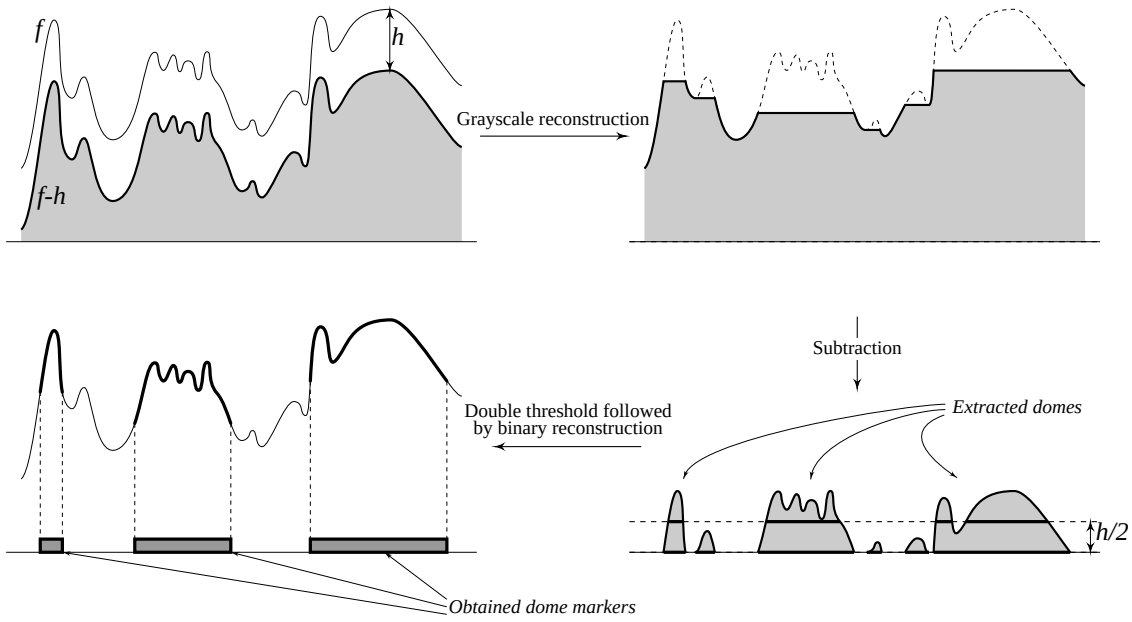


Figure 5 : *Algorithm used to extract the domes of a grayscale image.*

Here, the main problem consists in extracting reliable markers of the regions. The difficulty comes from the fact that very different images are to be used as input in our system. It is therefore impossible to make any assumptions on the shape and size of the regions as well as on the level of noise. One could think for example in using (like in section 2.3) the top-hat transformation to extract peaks and valleys. Unfortunately, this technique does not seem to be general enough, since it involves choosing some structuring elements, and therefore making an assumption on the shape and size of the regions to mark.

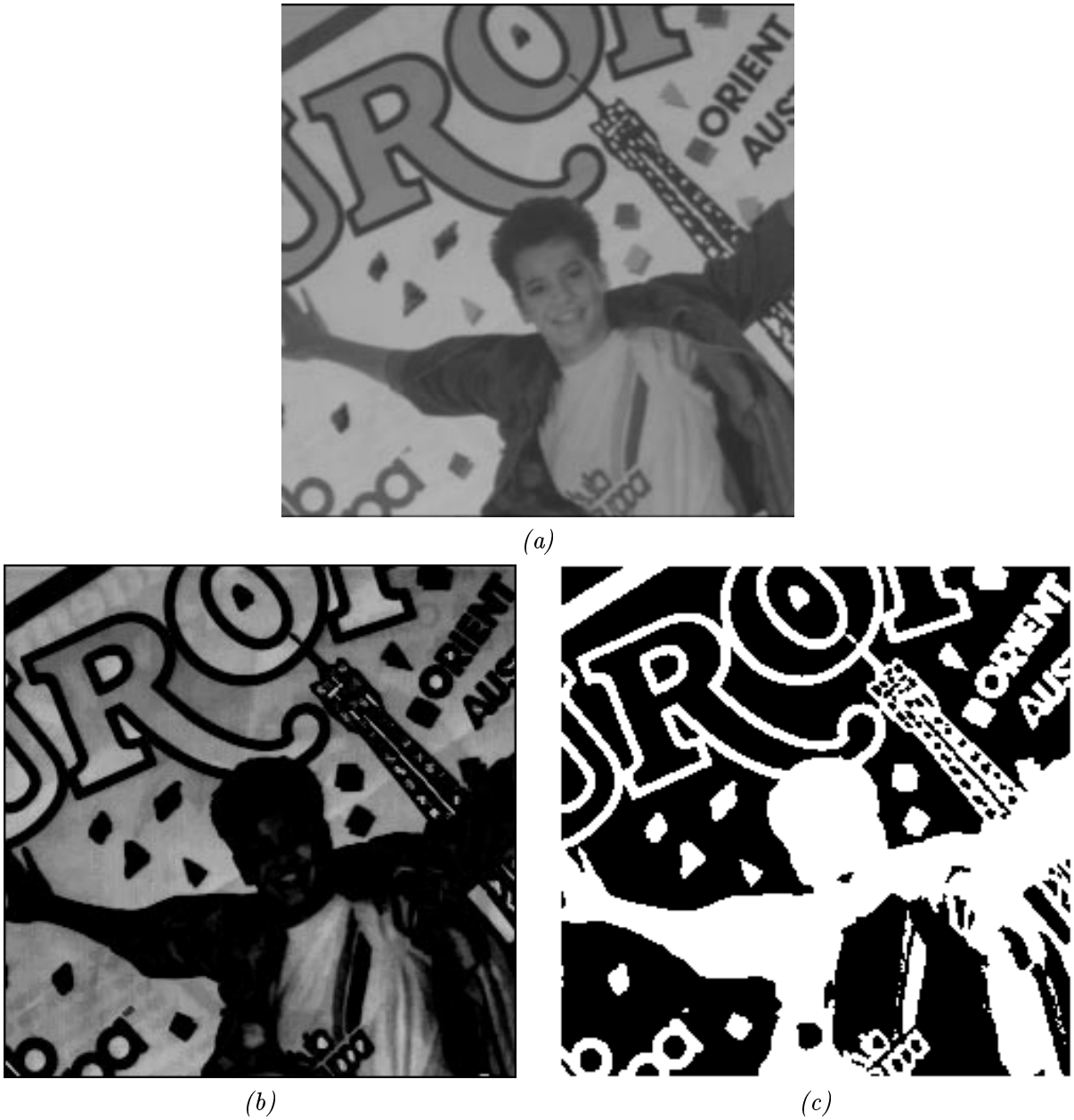
The method we finally used instead is sometimes called generalized maxima/minima extraction, or dome/basin extraction [46]. For the domes, e.g., the principle is to subtract an arbitrary constant h from the original image I and to perform a grayscale geodesic reconstruction of I from $I - h$ [44]. The grayscale reconstruction process can itself be viewed as an iteration of elementary dilations of $I - h$ with the constraint that at each step, the resulting image must be smaller than I for each pixel. The reconstructed image is then subtracted from the original one, thus yielding a grayscale image J of *all* the domes and crest-lines of I. ¿From J, it is then easy to extract a binary picture of the most important domes: it suffices to keep each dome which has at least one pixel with value greater than a given constant h'. Usually, one takes $h' = h/2$. This last operation is realized via binary reconstruction [17] of J thresholded at level 1 from J thresholded at value h'. The whole series of operations described in this paragraph is summarized by Figure 5. The dual process can be used to extract the basins and valleys of I.

One can observe that the above algorithm does not make any assumption on the shape or size on the regions. The only parameter it involves is the intensity constant h, which is related to the relative height of the extracted domes and crest-lines. In fact, the choice of h turns out to be not critical, since important variations of this parameter do not produce major changes of the extracted domes. Besides, one of the major advantages of watershed segmentation is that small variations of the shape of the markers have no influence on the final result.

The domes and basins obtained by applying this method to Figure 4.a with constant $h = 30$ gray levels are respectively shown in Figures 6 and 7. These results are then easily combined into one single marker image, shown in Figure 8.a. As explained, this marker image is used to modify the image gradient in Figure 4.b in order to impose on it these markers as new minima. Although this does not have a great influence on the result here, we chose to use a morphological gradient

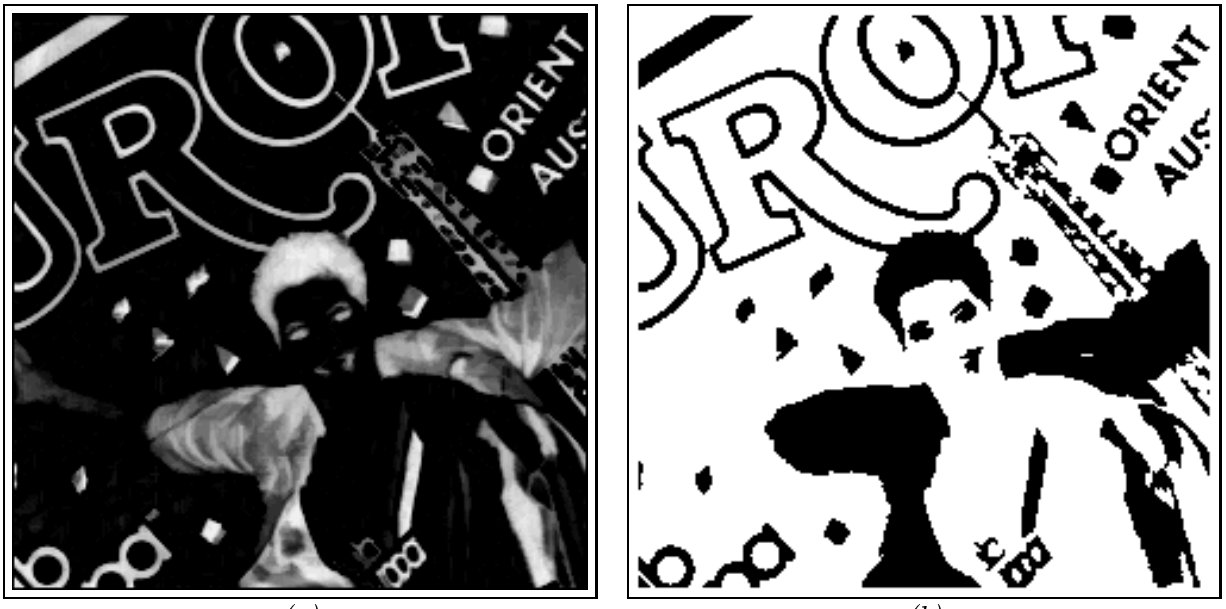
obtained as the maximum of 4 elementary directional gradients (i.e., gradients with respect to the 4 elementary line segments of the grid). The final segmentation is illustrated by Figure 8.b, superimposed to the original image.

It should be noted that this entire segmentation method is relatively fast. Indeed, the watershed algorithm used runs in approximately 1 or 2 seconds on a *Sun SparcStation 1,* for a 256×256-pixel image [46]. Similarly, the grayscale geodesic operations used twice in this procedure can be efficiently implemented either in a sequential fashion, or by using algorithms based on queues of pixels [43]. Their execution time is usually less than one second in the same conditions.



(a) *(b)* *(c)*

Figure 6 : *(a) Original image. (b) "Domes" of (a). (c) in black: markers of these domes, obtained after double threshold (see Figure 5).*



(a) *(b)*

Figure 7 : (a) "Basins" of Figure 6.a; (b) in black: markers of these basins.

(a) *(b)*

Figure 8 : (a) final marker image; (b) final segmentation of Figure 6.a.

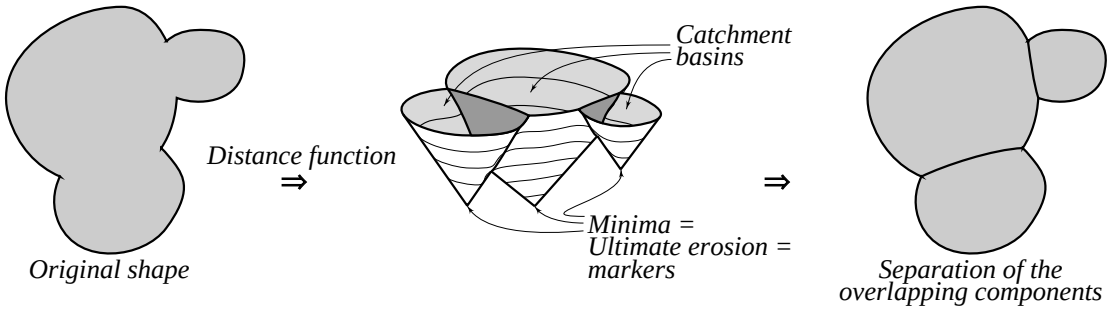


Figure 9 : *Binary segmentation by watersheds of the negative of the distance function.*

2.5 Watershed Segmentation Followed by Binary Region Segmentation

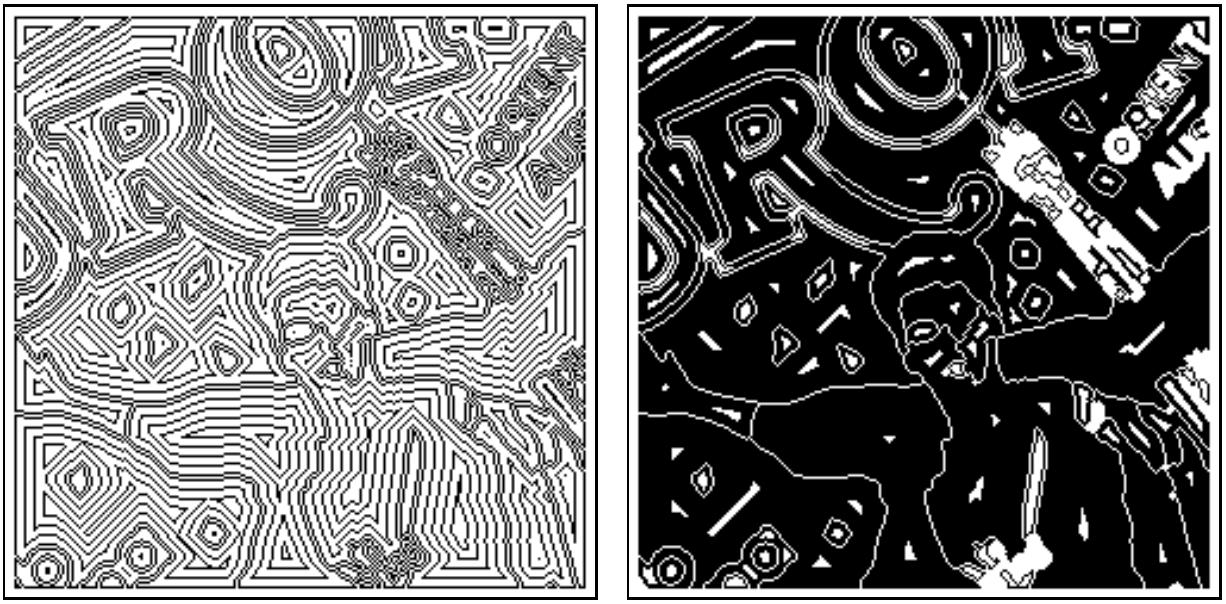
On Figure 8.b, one can notice that the obtained regions, though very accurate, sometimes exhibit very strange shapes. As we shall see later, this characteristics may have a bad effect on the matching algorithms, which usually work by using the centroid of the extracted regions. Furthermore, when the matching is done with few regions, the results are not as dense as one would expect, especially in comparison with the results provided by block matching techniques. Therefore, it is interesting at this point to cut the regions obtained after the above watershed segmentation into smaller pieces.

Although several approaches may be considered to achieve this goal, watershed-based methods seem once again to provide the most appropriate answer. We used here a technique which is commonly used for binary segmentation tasks, i.e., to separate binary shapes into their perceptually relevant *components.* This approach is detailed in [45, 46] and summarized in Figure 9. Its first step consists in determining the *distance function* of the binary image under study: each pixel belonging to the previously extracted regions is assigned a gray-level corresponding to its distance to the outer boundary of this region. Then, the maxima of such a distance function image are called *ultimate erosion* and mark the centroid of the different components in which the regions will be decomposed (see Figure 10). In actuality, in order to avoid getting too many markers, constrained maxima of the type presented in Figure 5 are used again here. Finally, the components are obtained by computing the catchment basins of the negation of the distance function, as illustrated by Figure 9. The result of this binary segmentation algorithm applied to the regions of Figure 8.a is shown in Figure 11.

2.6 Region Cleaning and Segmentation Results

The binary regions from the above four region extraction methods may be noisy. We clean them by first using a maximum of openings followed by a minimum of closings by 4 vector structuring elements oriented in four directions with length 3 pixels. This opening-closing filter eliminates noisy regions whose width in each of the four directions is smaller than the length of the structuring element, while preserving line-type regions, which usually convey important information. In addition, regions with areas less than a minimum of 9 pixels are considered too small to be reliable and hence they are eliminated. Hence, the overall region post-smoothing consists of a cascade of an opening-closing followed by an area filter.

Examples of regions resulting from the algorithms described in the previous subsections (including the pre- and post-smoothing) are presented in Figure 12. As Figure 12.b shows (with white areas representing the positive sign regions), the regions from the edge operator convey similar information as the edges. In contrast, the peak/valley regions in Figures 12.c and 12.d (where the peaks and valleys are represented by the white areas) correspond to intensity bright or dark blobs. Figure 12.e shows that watershed segmentation of the original gray-level image based on



(a) (b)

Figure 10 : *(a) Level lines of the distance function of the regions corresponding to Figure 8.b; (b) markers (in white) of the components in which the regions will be decomposed, superimposed to the regions themselves.*

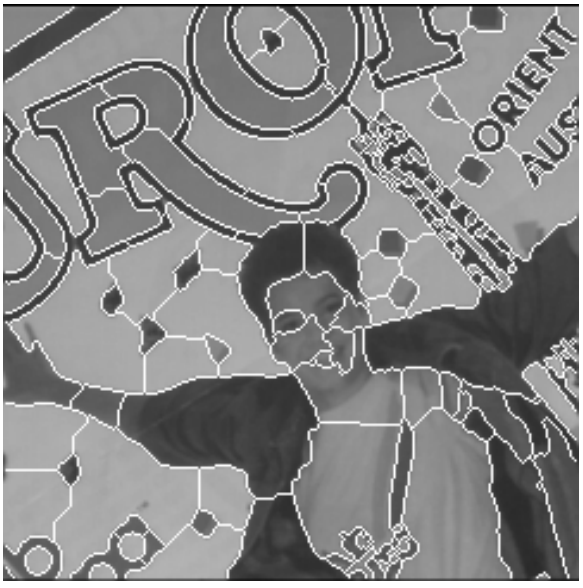


Figure 11 : *Final segmentation of the regions of Figure 8.b, superimposed to the original Figure 6.a*

dome/basin markers yields binary regions that are generally consistent with the concept of image segmentation. Finally, as shown in Figure 12.f, watershed segmentation of distance functions of binary regions resulting from graylevel watershed segmentation yields the densest region fields. (In Figures 12.e,f the region boundaries are overlapped to the original image.) The efficacy of these four region types for motion correspondence will be compared later.

3 Region Matching

3.1 Matching Criteria/Algorithm

Our region matching algorithm is guided by Ullman's general correspondence principles [41], but it also has two differences. First, the tokens Ullman used were 1-D line segments, whereas we use 2-D regions. Second, Ullman used affinity measures for matching, whereas we select the best region matching pair by comparing the similarities of regions based on an extended set of region features. Specifically, let R_i and R_j be two regions with areas $A(R_i)$ and $A(R_j)$, respectively, extracted from two consecutive image frames (at times $t = t_k, t_{k+1}$), and let $\vec{c_i}, \vec{c_j}$ denote their centroids. Then, fixing R_i, a region R_j from the frame at $t = t_{k+1}$ is a possible candidate to match with R_i if it successfully passes the following matching criteria:

1. *Centroid Distance:* For two centroids to match, their distance should not exceed an upper bound; i.e., both the x- and y-components of the displacement vector $\vec{c_i} - \vec{c_j}$ should not exceed L pixels.

2. *Region Identity*: The sign (positive or negative) of two regions if they resulted from the $\nabla^2 G * I$ approach, or their peak vs. valley (respectively, dome vs. basin) identities if they resulted from the morphological peak/valley (respectively, watershed segmentation) approach must be identical for allowing them to match.

3. *Area Difference*: The area of matching regions should not vary too much; i.e., $|A(R_i) - A(R_j)| < P \cdot A(R_i)$ where $0 < P < 1$.

4. *Intensity Difference:* The average intensities of the two regions should not vary too much; i.e.,
$$\left| \sum_{(x,y)\in R_i} \frac{I(x,y,t_k)}{A(R_i)} - \sum_{(x,y)\in R_j} \frac{I(x,y,t_{k+1})}{A(R_j)} \right| < ID_{max}.$$

Clearly, the fixed numbers L, P, and ID_{max} are control parameters for the correspondence process. Specifically, L controls the range of correspondence. Region R_i may be matched with R_j only if the centroid of R_j lies inside a square window of $(2L+1) \times (2L+1)$ pixels centered at the centroid of R_i, and their $\nabla^2 G * I$ signs or their peak/valley or dome/basin identities are the same. P and ID_{max} determine, respectively, the maximum percentage of area difference and the maximum average intensity difference between two regions above which a match is impossible. The parameters we used for these screening criteria in our experiments are $L = 25$ pixels, $P = 0.3$, and $ID_{max} = 20$.

If there are no regions R_j in the frame at $t = t_{k+1}$ satisfying the matching criteria, then there is no match for the particular region R_i. If more than one candidate regions R_j pass the matching criteria, the one having the smallest mean absolute intensity difference
$$\sum_{(x,y)\in R_i} |I(x,y,t_k) - I(x+d_x, y+d_y, t_{k+1})|$$
is selected, where $(d_x, d_y) = \vec{c_i} - \vec{c_j}$ is the centroid displacement vector. This final matching criterion has a similar effect as maximizing a nonlinear cross correlation consisting of a sum of minima [20].

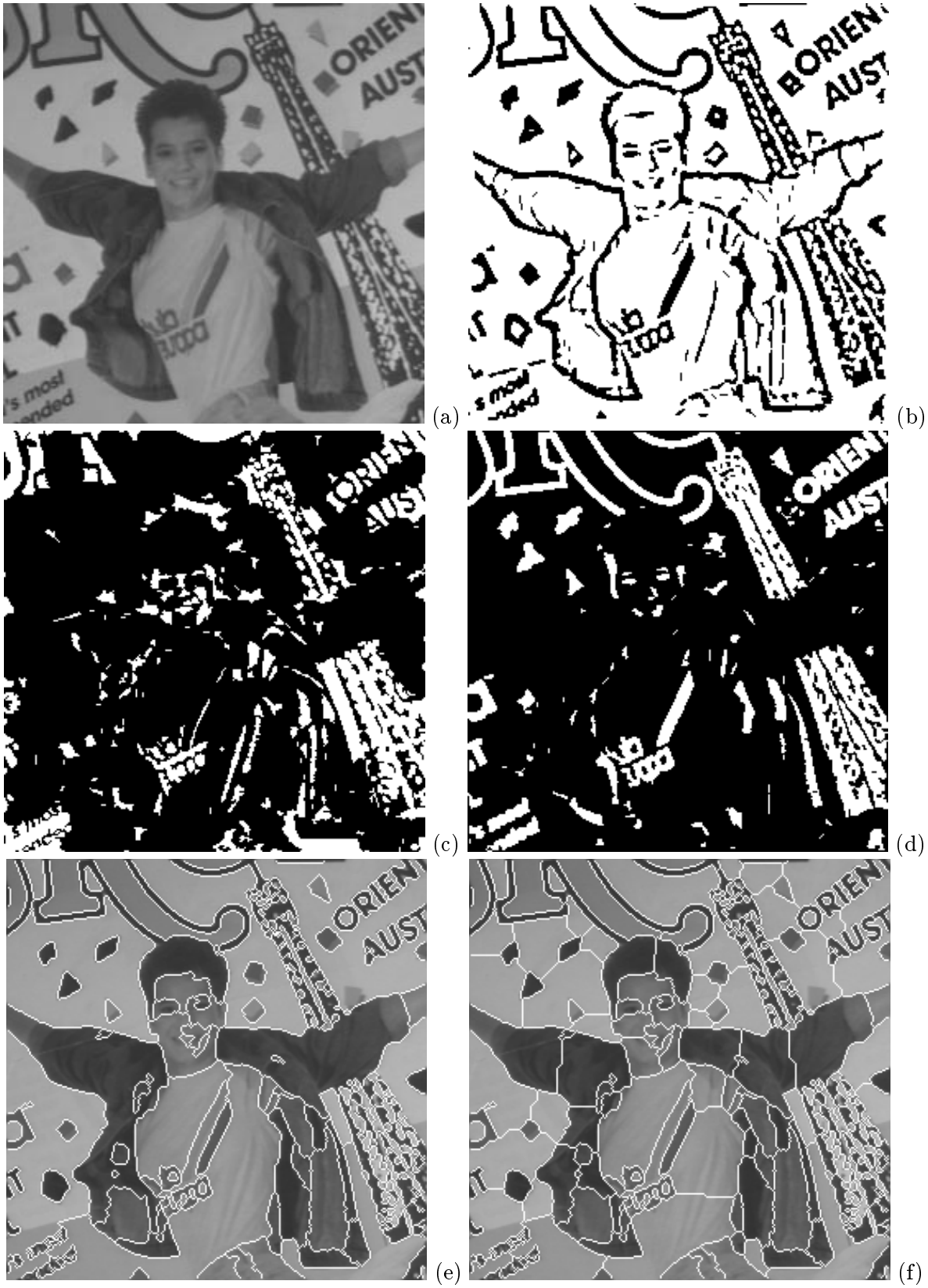


Figure 12 : *(a) Image I. Regions obtained by: (b) Sign of $\nabla^2 G * I$. (c) Peaks. (d) Valleys. (e) Watershed segmentation. (f) Watershed and binary region segmentation.*

This nonlinear correlation gives sharper matching peaks and is computationally faster than the linear (sum of products) correlation and its related mean squared matching error [20].

Each successful match of two regions in two consecutive image frames yields a spatial displacement vector (d_x, d_y) among the two region centroids. Estimating the velocity of a region's centroid by bringing it into correspondence with another region's centroid is not an arbitrary choice. The classical mechanics theory dictates that, with respect to an external force or torque, the motion of a rigid body can be represented by the motion of its centroid. Thus, we implicitly assume that each region is a small patch of a rigid body. We do not assume, however, that over a whole region the velocity remains constant. We simply estimate it only at the centroid. Finally, the average velocity is equal to $(v_x, v_y) = (d_x, d_y)/(t_{k+1} - t_k)$. Henceforth, we assume a uniform sampling of image frames in time and set $t_{k+1} - t_k = 1$, which amounts to equating pixel displacements with velocities.

Throughout the paper, images of the kind shown in Figure 13 are used as examples to illustrate the region matching procedures. These test images have been obtained by moving a camera at different positions in front of a poster-print image and digitizing the viewed image field. The problem is then to recover the apparent motion of the camera from the time series of image frames.

Figure 14 shows the result of matching the regions of Figures 13.a and 13.b extracted by the four algorithms described previously. Figure 15 provides similar results for the apparent motion between Figures 13.b and 13.c. In both cases, one can observe a few mismatches, mainly at the boundaries, but the overall displacement field seems reasonably correct. In particular, the rotation (first case) and zooming-out (second case) appear clearly. In these experiments velocity estimates were obtained up to 15-20 pixels in x and y directions.

To compare our region-based approaches for estimating 2-D image velocities we have simulated two other well known methods for 2-D motion detection: the block matching (also known as 'area correlation')[1] and the iterative gradient-based method of [12]. Figure 16 shows the result of the block matching and the gradient algorithm on the Poster image sequence. In general, the block matching method has computational complexity $O(B^2L^2G^2)$ where $B \times B$ are the pixel dimensions of the regions over which the error is averaged, L is the size of the search window for the optimum displacement, and $G \times G$ represents the pixel dimensions of the 2-D grid of locations at which we estimate displacement vectors. The complexity of the iterative gradient method [12] for $H \times W$-pixel image frames is about $O(HWN)$, where N is the number of required iterations for convergence. In the experiments reported in Figure 16 we used parameter values $B = 21$, $L = 51$, $G = 21$, $N = 256$. Note that the complexity of the edge-sign and peak/valley region extraction methods is about $O(HWK^2)$ where $K \times K$ are the pixel dimensions of the support of the Gaussian or opening-closing filters used for extracting edge-sign or peak/valley regions. The watershed segmentation can be done efficiently as in [46], and its computational complexity is linear in the number of pixels, i.e. $O(HW)$. The binary region segmentation uses the watershed segmentation of the distance function, and hence its computational complexity is also linear in the number of pixels. In addition, assuming that the screening criteria leave only few candidate regions for matching, the underlying mean absolute difference computation also has linear complexity in the number of pixels. Thus, overall, since G^2 is usually in the order of HW and K has the same order as B, due to the L^2 search factor the block matching method is computationally much more complex than the region-based approach. Hence, although the block matching gives satisfactory velocity estimates, it is very computation-intensive. Further, the standard block matching implicitly uses regions of

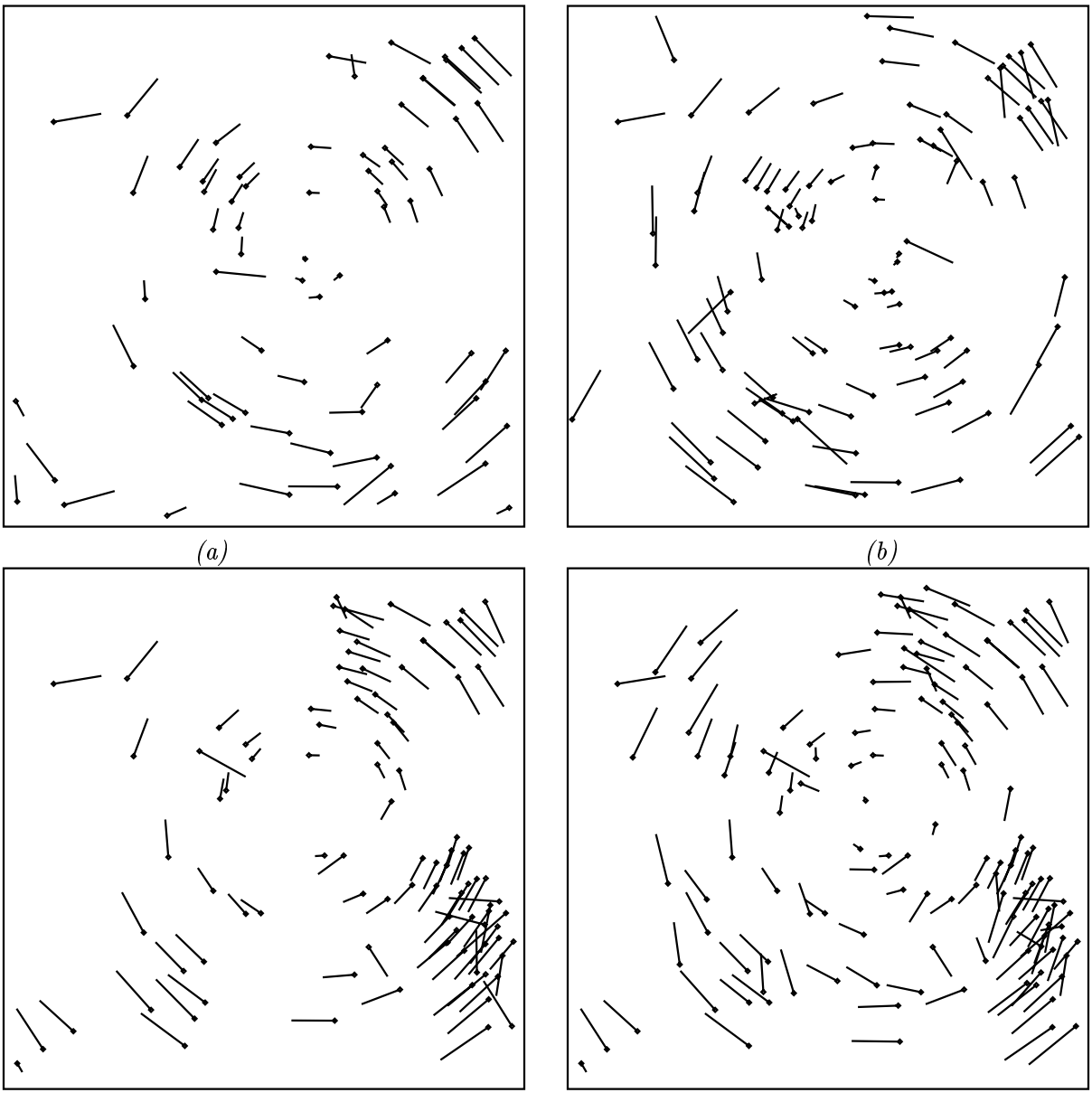
[1]The *block matching* is a well-known method, especially among researchers in video compression and remote sensing, to estimate 2-D velocities or pixel displacements on the image plane by minimizing $E(\vec{d}) = \sum_{\vec{p} \in R} |I(\vec{p}, t_1) - I(\vec{p} + \vec{d}, t_2)|^2$ over a small region R to find the optimum displacement vector $\vec{d}$. Minimizing $E(\vec{d})$ is closely related to finding $\vec{d}$ such that the correlation $\sum_{\vec{p} \in R} I(\vec{p}, t_1) I(\vec{p} + \vec{d}, t_2)$ is maximized; thus, it is sometimes called the *area correlation* method. A more efficient version of the method results from minimizing the mean absolute matching error.

(a)

(b)

(c)

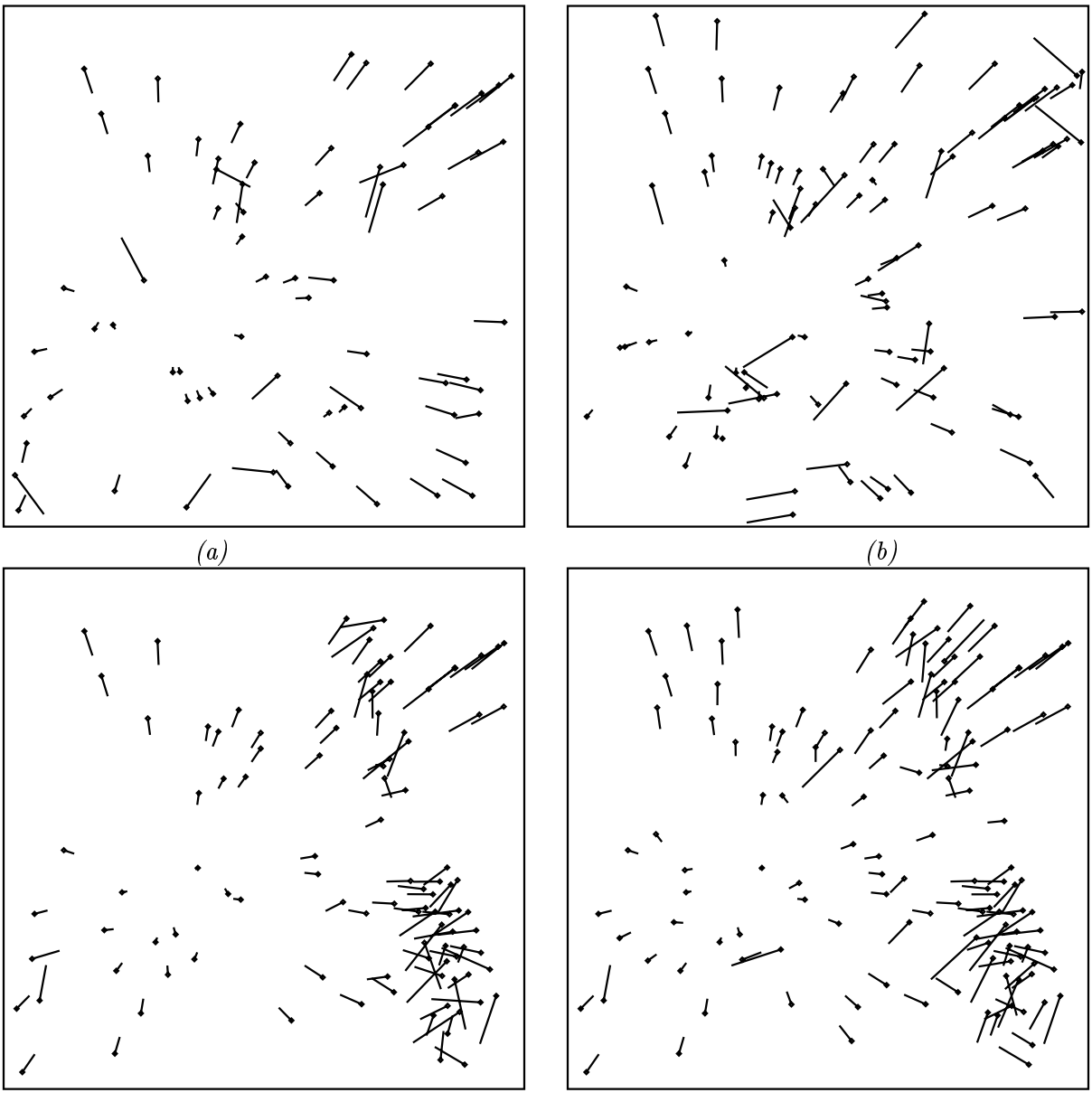
Figure 13 : *'Poster' sequence of test images (*256×256 *pixels). ¿From (a) to (b) the camera was rotated* 10° *counter clockwise, and from (b) to (c) it was translated along the camera's optical axis from a distance of 149 cm to a distance of 174 cm. (The camera axis was perpendicular to the poster object surface.)*



(a) *(b)*

(c) *(d)*

Figure 14 : *Velocity fields resulting from matching regions of Figures 13.a and 13.b extracted by: (a) Sign representation of* $\nabla^2 G * I$*; (b) Binarized peak/valleys; (c) Watershed segmentation; (d) Watershed segmentation followed by binary region segmentation.*



(a) *(b)*

(c) *(d)*

Figure 15 : *Velocity fields resulting from matching regions of Figures 13.b and 13.c extracted by: (a) Sign representation of* $\nabla^2 G * I$*; (b) Binarized peak/valleys; (c) Watershed segmentation; (d) Watershed segmentation followed by binary region segmentation.*

fixed size, whereas our approach allows for arbitrary regions and hence is much more flexible. On the other hand, as Figures 16.c,d show, the gradient algorithm has a very poor performance in detecting long-range motion; i.e., the pixel displacements ranged between 0 and 19 pixels. The gradient algorithm performs well only in very short-range motion, i.e., displacements by 1-2 pixels. In contrast, our region-based methods can provide good velocity estimates both for short- and long-range motion at a relatively low computational complexity.

3.2 Velocity Smoothing

Although many of the region matches appear to be accurate and robust, there may be a few mismatches. We view the mismatches as noise on the estimated velocity field. Then a question naturally arises of how to smooth the velocities.

We exclude the smoothing of the velocity vector field via linear filtering (e.g., local averaging) because linear smoothing filters have the well-known tendency to blur and shift sharp discontinuities in signals. In the case of velocity fields, these sharp discontinuities may indicate object boundaries and, hence, must be preserved. We choose median filtering because median filter is a nonlinear filter and can eliminate outliers or mismatches while preserve motion edges. Vector median filtering is defined to be the x, y componentwise median filtering:

$$med_i\{\vec{v_i}\} = (med_i\{v_{x,i}\}, med_i\{v_{y,i}\}).$$

where velocities $\vec{v_i} = (v_{x,i}, v_{y,i}), i = 1, 2, ..., n$ are the estimated velocities at various centroids around and including a centroid $\vec{c}$. Due to the relative sparseness of centroids, the estimates are found by searching inside a spatio-temporal cube centered at $\vec{c}$ and time t_k and whose size increases (but does not exceed twice the maximum window of matching) until n velocity estimates are found. In our experiments we set the parameter $n = 7$ and the size of the search cube 40 pixels in each spatial direction and three image frames along the time direction. Figure 17 illustrates the corresponding spatio-temporal vector median filtered results of Figure 15, which shows significant improvements. This vector median has been generally found to perform well in smoothing velocity fields [7]. (For a recent theoretical analysis of the vector median see [3].)

4 Experiments and Discussion

In this section we provide some empirical numerical results from experiments on comparing the estimation error of the four region-based approaches applied either on clean images or on images corrupted by adding salt-and-pepper or white Gaussian noise. While our numerical results refer only to the Poster image sequence of Figure 13 for purposes of brievity of exposition, we have reached similar conclusions by applying our algorithms to a large variety of moving images of outdoors and indoors scenes.

To simulate motion by a known amount in our numerical comparisons, the image of Figure 13.a was translated vertically and horizontally by the same number of pixels. Then both the original and the translated images were corrupted with salt-and-pepper or white Gaussian noise. Figure 18 shows the image of Figure 13.a corrupted with 5% and 10% salt-and-pepper noise, as well as with white Gaussian noise at levels of 30 and 20 dB of peak-to-peak signal-to-noise-ratio (SNR). The SNR is defined as $20\log_{10}(255/\sigma_n)$, where σ_n is the standard deviation of the noise. Tables 1,2 and 3,4 show the performance of the four region-based approaches followed by median smoothing of the velocities in estimating the 2-D motion between original and translated in the cases of 10×10 and 20×20 pixel translations. Specifically, these Tables tabulate the average estimation error in both the x- and y-component of the displacement vector, averaged over all regions to which a velocity was assigned. They also show the number of regions in the first image frame, which were successfully matched and assigned a velocity vector.

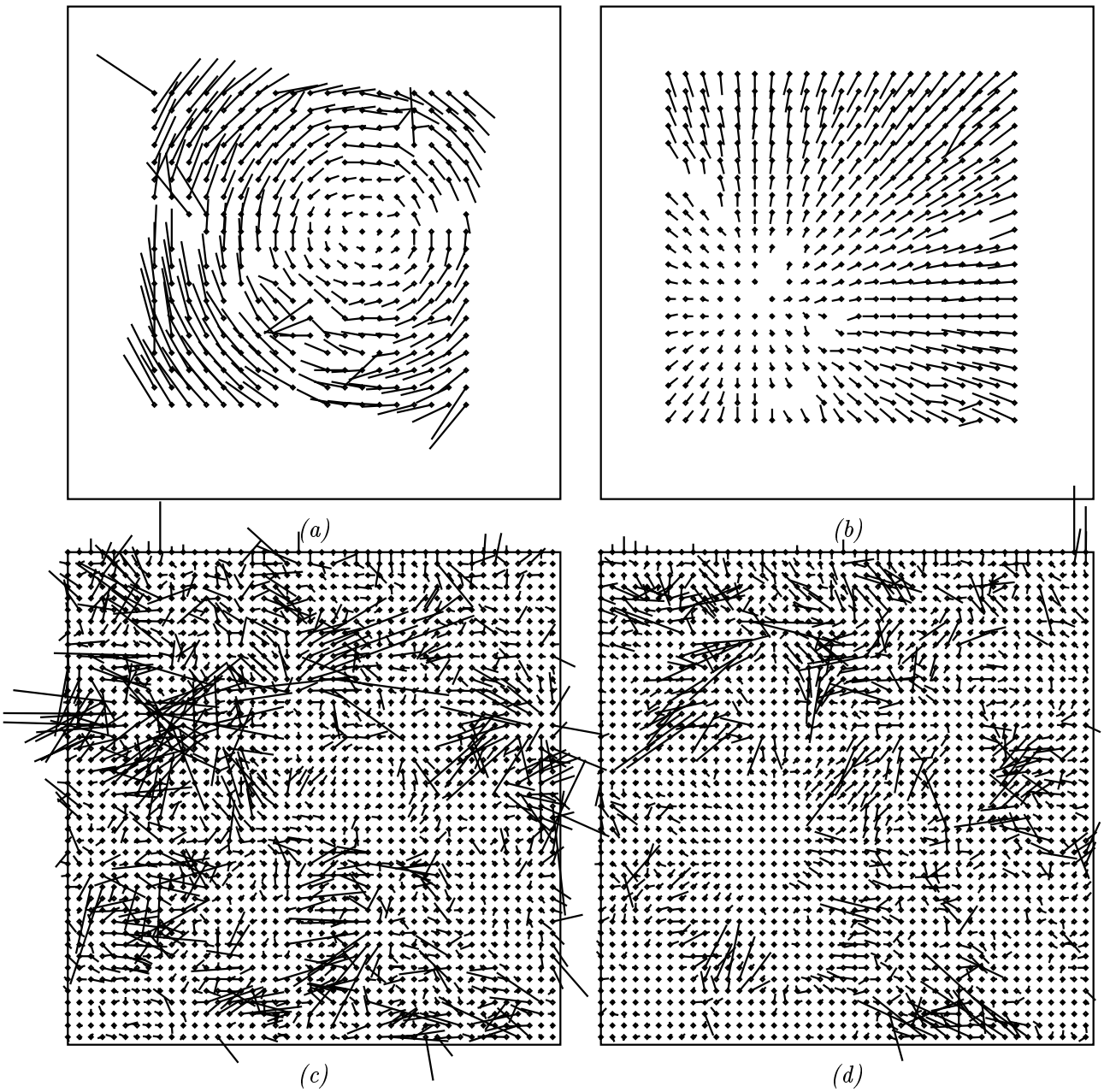


Figure 16 : *Velocity fields resulting from: (a) Block matching of the images in Figures 13.a and 13.b. (b) Block matching applied to Figures 13.b and 13.c. (c) Gradient algorithm [12] applied to Figures 13.a and 13.b. (d) Gradient algorithm [12] applied to Figures 13.b and 13.c.*

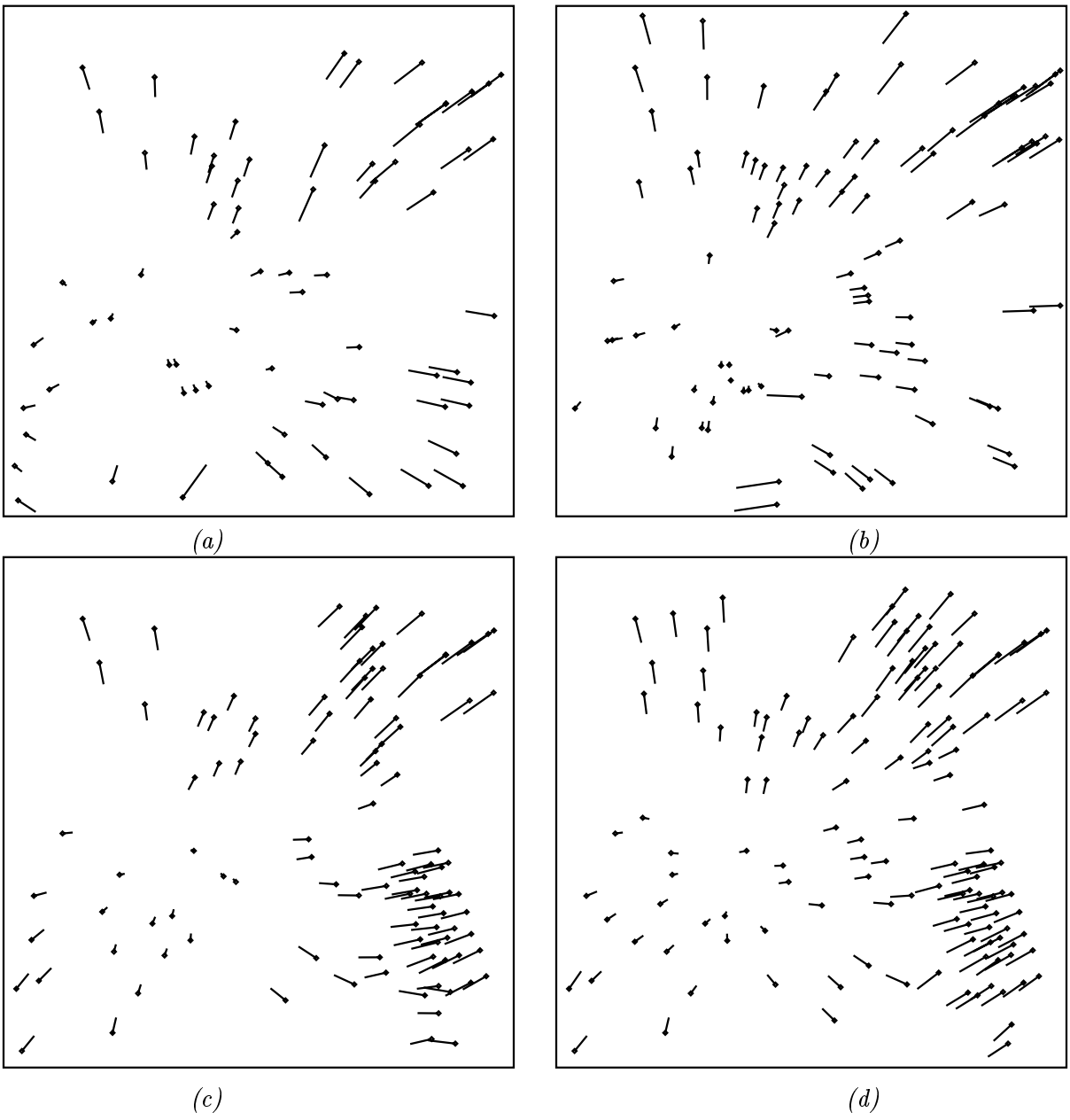


Figure 17: *Velocity fields of Figure 15 smoothed via a spatio-temporal vector median filter.*

Table 1 : *Translation* 10×10 *pixels, Salt-and-Pepper Noise*

	No noise			5% noise			10% noise		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	0.0082	0.0012	85	0.0788	0.1390	73	0.2161	0.4339	59
Peak/Valley	0.0713	0.1327	127	0.1277	0.1164	110	0.2125	0.2807	88
Watershed	0.0000	0.0018	114	1.3960	0.7715	100	1.9734	2.1661	96
Wat.&Bin.Seg.	0.0006	0.0015	162	0.9581	0.9822	135	2.3427	2.2573	137

Table 2 : *Translation* 10×10 *pixels, White Gaussian Noise*

	No noise			SNR=30 dB			SNR=20 dB		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	0.0082	0.0012	85	0.7082	0.4352	61	2.8658	2.0162	136
Peak/Valley	0.0713	0.1327	127	0.6473	0.7132	91	1.7753	1.6258	99
Watershed	0.0000	0.0018	114	1.4331	0.8904	266	4.9306	1.7426	417
Wat.&Bin.Seg.	0.0006	0.0015	162	1.3203	0.8198	303	4.7689	1.7307	417

Table 3 : *Translation* 20×20 *pixels, Salt-and-Pepper Noise*

	No noise			5% noise			10% noise		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	0.6408	0.2487	76	0.3400	0.8164	70	1.1794	1.2183	63
Peak/Valley	0.0438	0.3223	121	0.3103	0.2848	102	1.2506	0.8698	86
Watershed	0.0032	0.0000	95	1.9323	0.9167	96	5.7756	3.9074	129
Wat.&Bin.Seg.	0.0504	0.0011	139	5.8425	1.6827	127	4.2118	3.9048	165

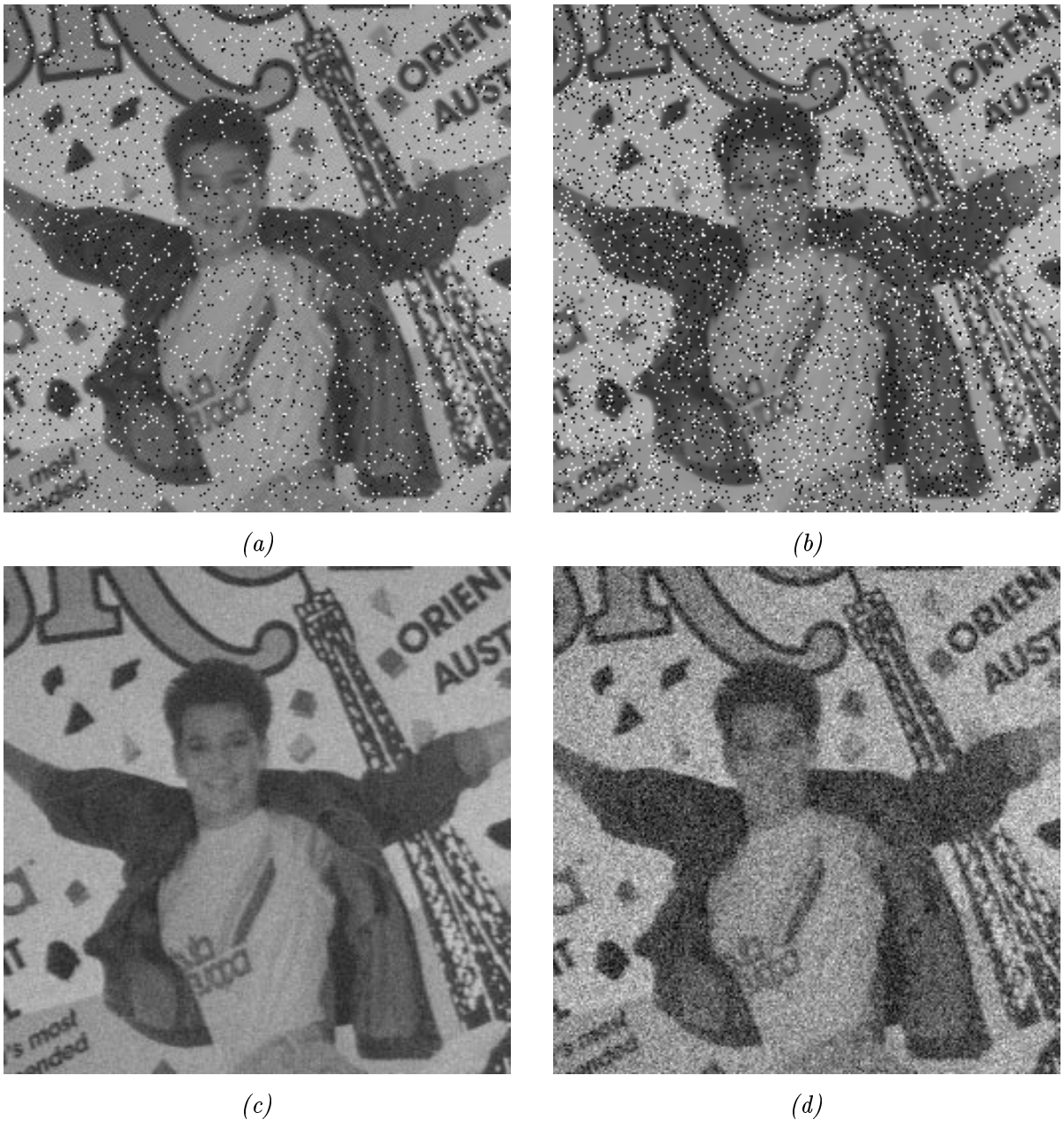
Table 4 : *Translation* 20×20 *pixels, White Gaussian Noise*

	No noise			SNR=30 dB			SNR=20 dB		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	0.6408	0.2487	76	0.2781	0.6555	64	5.3332	2.6679	131
Peak/Valley	0.0438	0.3223	121	2.7898	2.0333	93	4.9663	5.6870	104
Watershed	0.0032	0.0000	95	3.8875	1.6351	276	7.0879	2.5315	459
Wat.&Bin.Seg.	0.0504	0.0011	139	3.2130	1.9092	319	7.2363	2.2531	463

As Tables 1,2,3,4 show, when there is no noise,[2] all four approaches perform very well in estimating translations of 10 and 20 pixels in each direction, since the error is a small or negligible fraction of a pixel. This small amount of error is caused by the image boundaries where regions appear or disappear. In the noise-free case, the watershed segmentation followed by binary region segmentation yields very small errors and the densest velocity estimates. However, the two watershed segmentation approaches (with or without binary region segmentation) are more sensitive to noise than the edge-sign or peak/valley region approaches. Hence, the watershed with binary region segmentation is the most recommended approach when the noise level is very low.

In the presence of white Gaussian or salt-and-pepper noise, the $\nabla^2 G$ edge-sign and the morphological peak/valley region approaches perform similarly and better than the watershed approaches. Specifically for 5%–10% salt-and-pepper and for 30 dB white Gaussian noise both approaches yield a translation estimation error in the order of about 1–10%. For 20 dB white Gaussian noise, this error order increases to 15%–25%. In some of our past experiments that did not include image pre-

[2]In the noise-free case, the regions of the original and artificially translated image should be the same, except for these close to the boundary, because the image smoothing, region extraction, and region cleaning algorithms are all translation-invariant. Hence, in this case all the displacement estimates from our algorithms should be correct except at boundaries; our experiments confirmed this fact.



(a) *(b)*

(c) *(d)*

Figure 18 : *The image of Figure 13.a corrupted by: (a) 5% salt-and-pepper noise; (b) 10% salt-and-pepper noise; (c) white Gaussian noise, SNR=30 dB; (d) white Gaussian noise, SNR=20 dB.*

Table 5 : *Rotation $\theta = 5^o$, Salt-and-Pepper Noise*

	No noise			5% noise			10% noise		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	0.6793	0.6821	80	0.6924	0.6587	76	0.7998	1.1892	69
Peak/Valley	0.6544	0.6757	115	0.6866	0.9265	94	1.0953	1.1696	84
Watershed	1.0773	0.7163	102	1.5338	1.2532	111	2.2324	1.4570	120
Wat.&Bin.Seg.	0.9558	0.8951	129	1.3837	1.5574	142	2.0657	1.8908	150

Table 6 : *Rotation $\theta = 5^o$, White Gaussian Noise*

	No noise			SNR=30 dB			SNR=20 dB		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	0.6793	0.6821	80	0.8304	0.7589	62	2.2189	3.5590	133
Peak/Valley	0.6544	0.6757	115	0.9501	1.1627	89	1.9477	2.2813	101
Watershed	1.0773	0.7163	102	1.1765	3.0664	272	2.3182	5.0146	412
Wat.&Bin.Seg.	0.9558	0.8951	129	1.2113	2.9663	311	2.4222	5.0465	416

Table 7 : *Rotation $\theta = 10^o$, Salt-and-Pepper Noise*

	No noise			5% noise			10% noise		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	1.6475	1.2946	77	1.8442	1.4170	72	2.4996	3.2853	62
Peak/Valley	1.6383	1.4963	110	1.7134	1.5889	96	2.5090	1.5844	92
Watershed	1.9134	1.3308	101	2.3950	3.1793	119	3.9043	4.5042	134
Wat.&Bin.Seg.	1.6664	1.5161	135	3.0048	3.7086	162	3.6614	5.3904	170

Table 8 : *Rotation $\theta = 10^o$, White Gaussian Noise*

	No noise			SNR=30 dB			SNR=20 dB		
	x err	y err	vel#	x err	y err	vel#	x err	y err	vel#
$\nabla^2 G$ edge sign	1.6475	1.2946	77	2.0777	2.3999	69	3.1877	3.9804	164
Peak/Valley	1.6383	1.4963	110	1.6820	1.7981	89	5.0176	4.3769	102
Watershed	1.9134	1.3308	101	3.0208	6.5733	304	4.1037	7.7669	476
Wat.&Bin.Seg.	1.6664	1.5161	135	3.0777	6.3587	341	4.0219	7.7885	482

smoothing and velocity post-smoothing we had observed that the $\nabla^2 G$ edge sign performed better that the peak/valley approach in white Gaussian noise, whereas the opening-closing peak/valley approach performed better in salt-and-pepper noise. This is somewhat expected because the linear Gaussian smoother performs better in suppressing white Gaussian noise, whereas the nonlinear opening-closing smoothers are superior for suppressing salt-and-pepper noise. However, in our present system the image pre-smoothing via alternating sequential filtering and the velocity post-smoothing via vector median filtering tend to blur the distinctions between the edge sign and peak/valley approaches. One difference, however, is that the peak/valley approach tends to yield denser velocity fields than the edge-sign approach in most cases.

Tables 5,6,7,8 show the average displacement estimation errors and the number of velocity estimates using the four region approaches for a simulated rotational motion of the image in Figure 13.a by amounts of $\theta = 5^o$ and 10^o degrees, in a noise-free case as well as in the presence of noise. Note that the rotations by 5^o and 10^o with respect to the image center incur displacements ranging between 0–11 and 0-22 pixels. As Tables 5,6,7,8 show, the average displacement estimation errors were in the order of 1-2 pixels for all four methods in the noise-free case. In the noisy case (5%–10% salt-and-pepper noise and 30-20 dB white Gaussian noise), the edge-sign and peak/valley approaches yielded average errors in the order of 1–5 pixels, whereas the errors of the watershed

approaches were in the order of 1–8 pixels. ¿From the estimation error viewpoint, it appears that all four approaches perform similarly in the noise-free rotation case, since they yield average errors of similar order. The watershed followed by binary segmentation yields again the densest velocity fields. In the noisy case, the edge-sign and peak/valley approaches perform better than the watershed approaches.

5 Conclusion

We have developed four region-based approaches to solve the visual motion correspondence problem and compared their performance. The main part of the work is the region extraction phase. Thus, in each image frame the regions are extracted from any of the four following approaches: (i) the sign representation of the $\nabla^2 G$ edge operator; (ii) thresholding morphological peak/valley transformations; (iii) morphological watershed segmentation of image gradients using dome/basin markers; (iv) watershed segmentation of image gradient followed by watershed segmentation of the distance functions of the resulting binary regions. The motion correspondence problem is then solved by matching the extracted regions via a procedure that compares regions based on similarities among several of their features. Image velocities are identified as the spatial displacement vectors between centroids of corresponding regions. The 2-D velocity estimates are then smoothed by a spatio-temporal vector median filter.

Several numerical comparisons have been done in the absence or presence of noise to compare the four region approaches. The results indicate that in the noise-free case the watershed followed by binary region segmentation is the best approach in yielding one of the smallest (in translational motion) or similar-order (in rotational motion) estimation errors and the densest velocity fields. In the noisy case both the edge-sign and peak/valley approaches perform similarly and better than the watersheds, with a small advantage of the peak/valley approach in giving denser velocity fields than the edge-sign approach.

Our experiments on real and synthetic moving imagery provide strong evidence that the developed region-based system for 2-D displacement estimation performs well for both short- and long-range motion and has some advantages over two other approaches. Specifically, while the region-based approach has similar performance with the block matching method, the latter is computationally much more intense. In addition, for medium- or long-range motion the region-based approach performed much better than the iterative gradient method of [12].

Finally, in addition to their usefulness for motion tracking, the developed morphological region extraction methods can also serve as efficient systems for robust 2-D feature extraction in a variety of computer vision tasks.

References

[1] J.K. Aggarwal, L.S. Davis and W.N. Martin, "Correspondence Processes in Dynamic Scene Analysis", *Proc. IEEE,* 69, pp. 562-572, May 1981.

[2] J.K. Aggarwal and N. Nandhakumar, "On the computation of Motion from Sequences of Images–A Review", *Proc. IEEE, 76,* pp.917-935, Aug. 1988.

[3] J. Astola, P. Haavisto, and Y. Neuvo, "Vector Median Filters", *Proc. IEEE*, 78, pp. 678-689, April 1990.

[4] S.T. Barnard and W.B. Thompson, "Disparity analysis in images", *IEEE Trans. Pattern Anal. Mach. Intell.* 2, pp. 333-340, 1980.

[5] S. Beucher & Ch. Lantuéjoul, "Use of Watersheds in Contour Detection", *Proc. International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation,* Rennes, France, 1979.

[6] R. Brockett, "Gramians, Generalized Inverses, and the Least-Squares Approximation of Optical Flow", *J. Visual Commun. Image Repres.*, 1, pp.3-11, Sep. 1990.

[7] C.S. Fuh and P. Maragos, "Region-Based Optical Flow Estimation", *IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, June 1989.

[8] C.S. Fuh and P. Maragos, "Application of Mathematical Morphology to Motion Image Analysis", in *Proc. Electronic Imaging EAST Conference*, Boston, Oct. 1990.

[9] E. C. Hildreth, "Computations underlying the measurement of visual motion", *Artif. Intellig.*, 23, pp. 309-354, 1984.

[10] E. C. Hildreth and C. Koch, "The Analysis of Visual Motion: From Computational Theory to Neuronal Mechanisms", A.I.L. Memo 919, M.I.T., 1986.

[11] B. K.P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.

[12] B.K.P. Horn, and B.G. Schunck, "Determining Optical Flow", *Artificial Intelligence,* vol. 17, nos. 1-3, pp. 185-203, August, 1981

[13] T.S. Huang and R.Y. Tsai, "Image Sequence Analysis: Motion Estimation", in *Image Sequence Analysis,* T.S. Huang, Ed., Springer-Verlag, 1981.

[14] J.R. Jain and A.K. Jain, "Displacement Measurement and Its Application in Interframe Coding," *IEEE Trans. Commun.,* COM-29, pp. 1799-1808, Dec. 1981.

[15] K. I. Kanatani, "Transformation of Optical Flow by Camera Rotation" *IEEE Trans. Pattern Anal. Mach. Intellig.*, PAMI-10, Mar. 1988.

[16] J.J. Koenderinck and A.J. van Doorn, "Local structure and movement parallax of the plane", *J. Opt. Soc. Amer.,* 66, pp. 717-723, July 1976.

[17] Ch. Lantuéjoul and F. Maisonneuve, "Geodesic Methods in Image Analysis", *Pattern Recognition,* Vol. 17, pp. 117–187, 1984.

[18] J. S. J. Lee, R. M. Haralick, and L. G. Shapiro, "Morphologic Edge Detection", *IEEE Trans. Rob. Autom.*, vol. RA-3, pp. 142-156, Apr. 1987.

[19] H. C. Longuet-Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image", *Proc. Roy. Soc. London*, B 208, pp. 385-397, 1980.

[20] P. Maragos, "Morphological Correlation and Mean Absolute Error Criteria", in *Proc. Int'l Conf. Acoust. Speech and Signal Processing*, Glasgow, Scotland, May 1989.

[21] P. Maragos and R.W. Schafer, "Morphological Filters", *IEEE Trans. Acoust. Speech Signal Process.,* ASSP-35, Aug. 1987.

[22] P. Maragos and R.W. Schafer, "Morphological Systems", *Proc. IEEE*, 78, pp. 690-710, April 1990.

[23] D. Marr, *Vision*, W.H. Freeman & Co., San Francisco, 1982.

[24] D. Marr and E.C. Hildreth, "Theory of edge detection", *Proc. Roy. Soc. Lond. B,* 207, pp.187-217, 1980.

[25] J. E.W. Mayhew and J. P. Frisby, "Psychophysical and Computational Studies towards a Theory of Human Stereopsis", *Artificial Intelligence*, 17, pp.349-385, 1981.

[26] F. Meyer, "Contrast Feature Extraction", in *Special Issues of Practical Metallography*, Stuttgart, Germany: Riederer Verlag, GmbH, 1978. Also in *Proc. 2nd European Symp. Quantitative Analysis of Microstruct. in Materials Science, Biology, and Medicine*, France, Oct. 1977.

[27] F. Meyer and S. Beucher, "Morphological Segmentation", *J. Visual Commun. and Image Representation*, vol. 1, pp. 21-46, Sep. 1990.

[28] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in Picture Coding", *Proc. IEEE*, 73, pp. 523-548, 1985.

[29] A.N. Netravali and J.D. Robbins, "Motion compensated television coding– Part I," *Bell Syst. Tech. J.,* 58, pp. 631-670, March 1979.

[30] H. K. Nishihara, "Practical real-time imaging stereo matcher", *Optic. Enginr.,* 23(5), pp.536-545, 1984.

[31] V. S. Ramachandran and S. M. Anstis, "The Perception of Apparent Motion", *Scientific American*, pp. 102-109, June 1986.

[32] A. Rosenfeld and J.L. Pfaltz, "Distance Functions on Digital Pictures", *Pattern Recognition,* Vol. 1, pp. 33–61, 1968,

[33] F. Safa and G. Flouzat "Speckle Removal on Radar Imagery Based on Mathematical Morphology", *Signal Processing*, 16, pp. 319-333. 1989.

[34] D. Schonfeld and J. Goutsias, "Optimal Morphological Pattern Restoration from Noisy Binary Images", *IEEE Trans. Pattern Anal. and Machine Intellig.*, PAMI-13, pp. 14-29, Jan. 1991.

[35] J. Serra, *Image Analysis and Mathematical Morphology,* Academic Press, London, 1982.

[36] J. Serra (ed.), *Image Analysis and Mathematical Morphology. Vol 2: Theoretical Advances,* Academic Press, London, 1988.

[37] J. Serra and L. Vincent, "An overview of morphological filtering", to appear in *Circuits, Systems and Signal Processing,* 1991.

[38] S. R. Sternberg, "Grayscale Morphology", *Computer Vision, Graphics and Image Processing*, 35, pp. 333–355, 1986.

[39] R. L. Stevenson and G. R. Arce, "Morphological Filters: Statistics and Further Syntactic Properties", *IEEE Trans. Circ. and Syst.*, CAS-34, pp.1292-1305, Nov. 1987.

[40] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces", *IEEE Trans. on Patt. Anal. Mach. Intell.,* vol. 6, pp. 13-27, Jan. 1984.

[41] S. Ullman, *The Interpretation of Visual Motion,* MIT press, 1979.

[42] L.J. Van Vliet and I.T. Young, "A Nonlinear Laplace Operator as Edge Detector in Noisy Images", *Computer Vision, Graphics, and Image Processing,* vol. 45, pp. 167-195, 1989.

[43] L. Vincent, "New Trends in Morphological Algorithms", *Proc. SPIE/SPSE Vol. 1451, Nonlinear Image Processing II,* pp. 158–170, San Jose (CA), February 1991.

[44] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Efficient Algorithms and Applications", Tech. Report 91-16, Harvard Robotics Lab., 1991.

[45] L. Vincent and S. Beucher, "The morphological approach to segmentation: an introduction", *Technical Rep. CMM, Ecole Nationale Supérieure des Mines de Paris,* France, 1989.

[46] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 13 (6), pp. 583–598, 1991.

[47] A.M. Waxman, and S. Ullman, "Surface Structure and 3-D Motion From Image Flow: A Kinematic Analysis", *Intl. Journal of Robotics Research* 4, pp 72-94, 1985.

[48] A.M. Waxman and K. Wohn, "Image Flow Theory: A Framework for 3-D Inference from Time-Varying Imagery", in *Advances in Computer Vision,* Vol. 1, C. Brown, Ed., NJ:Erlbaum Publ., 1988.

[49] J. Weng, T.S. Huang, and N. Ahuja, "Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation", *IEEE Trans. on Patt. Anal. Mach. Intell.,* vol. 11, pp. 451-476, May 1989.

[50] J. Wu, R. Brockett and K. Wohn, "A Contour-based Recovery of Image Flow: Iterative Method", *IEEE Conf. on Computer Vision and Pattern Recognition,* pp. 124-129, San Diego, June 1989.