Motion Segmentation and Depth Ordering Based on Morphological Segmentation

Lothar Bergen and Fernand Meyer

bergen@cmm.ensmp.fr, meyer@cmm.ensmp.fr,
Centre de Morphologie Mathématique, Ecole des Mines de Paris,
35, rue Saint-Honoré, 77305 Fontainebleau Cedex, France

Abstract. In this paper the motion segmentation and depth ordering problem for monocular image sequences with and without camera motion is addressed. We show how a new multiscale morphological segmentation technique, based on the watershed, can produce a superset of the motion boundaries. Regions with similar motion then have to be merged. The difficulties of motion estimation at object boundaries with occlusion are analyzed and a solution combining segmentation and robust estimation is presented. Region merging is then performed using the obtained motion parameters. We then present a new technique for the depth ordering of the resulting image partition. We show how the modelling error on either side of the motion boundary can be used to indicate the occlusion relationship of the objects. The algorithm is then applied to several synthetic and natural image sequences. The results demonstrate that the technique is robust and that the depth ordering requires only minimal motion to perform correctly. This is due to the fact that, unlike existing techniques for depth ordering, the motion between two frames only has to be analyzed. We then point out possible improvements and indicate how temporal integration of the information can further increase stability.

1 Introduction

The increasing availability of audiovisual material in digital form creates a demand for new functionalities like interactivity, integration of objects of different nature, etc.. The new standard MPEG-4 meets these demands by allowing a scene to be represented as a composition of objects rather than just pixels. It does not specify, however, how the decomposition of a scene into objects is performed.

A first step in the semantic analysis of a scene is the segmentation into objects with coherent motion. A second step then consists in establishing the depth ordering of the resulting image partition: to establish which object moves in front of which.

For the motion segmentation two possible starting points exist.

We can start with a motion field we want to segment: in this case, the field needs to be dense and accurate. Motion estimation unfortunately produces poor results precisely at motion boundaries.

The use of grey level segmentation is the alternative starting point. The hypothesis underlying this approach is that such a segmentation produces a superset of the motion boundaries: the motion boundaries are contained in the segmentation. The problem then consists in merging regions with similar motion, which also proves challenging at object boundaries with occlusion.

The techniques that perform motion segmentation and depth ordering found in literature rely exclusively on motion information: some approaches try to detect motion boundaries directly in sparse motion fields calculated through token matching [6, 7]. A more recent technique is based on a decomposition of the scene into layers with coherent motion. The evolution of these layers in time is then used to extract information concerning depth ordering [2].

In this paper, we address the motion segmentation and depth ordering problem for monocular image sequences with and without camera motion. We use a morphological grey level segmentation as our starting point. We then show how robust parameter estimation techniques improve motion estimation at motion boundaries and how this permits regions with similar motion to be merged. A new technique for the depth ordering of the resulting image partition is then presented.

Figure 1 shows a schematic overview of our algorithm which also corresponds to the structure of this article.

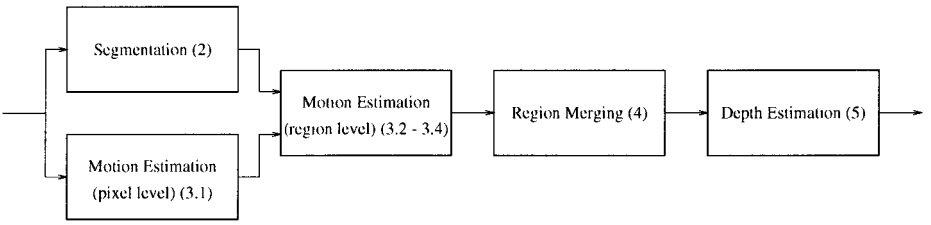


Fig. 1. The steps of the motion segmentation and depth ordering

2 Segmentation

Segmentation generally produces a superset of the motion boundaries: the motion boundaries are included in the grey level segmentation. This is due to the fact that the surface properties or the illumination of the objects in the scene often differ.

Figure 2 shows, as an example, image no. 50 from the "Foreman" sequence in QCIF format. Next to the original we see the morphological gradient, the difference between the dilated and eroded image, which indicates discontinuities in the luminance (shown with $\gamma = 5$ for better visualization).

We can confirm that the motion boundary (the contour of the upper body) corresponds to areas with high gradient almost everywhere.

The segmentation we have used for this work is a multiscale morphological segmentation technique, based on volumic closings of the gradient image and the watershed transform [1]. The result is a series of mosaic images with increasing

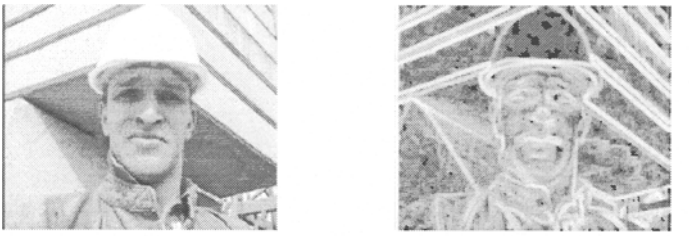


Fig. 2. "Foreman"

resolution verifying the following property: each contour present in a given mosaic is also present in all finer mosaics. Figure 3 shows such a series of mosaics for the above example.

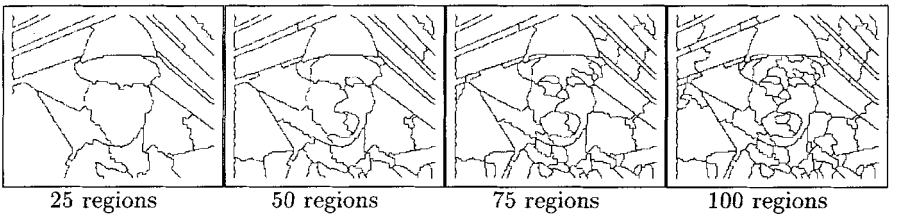


Fig. 3. Hierarchy of segmentations

This segmentation is extremely fast, since all resolution levels are constructed simultaneously in the same run. Hence, it is easy to choose the best starting point for studying the motion.

Two basic strategies are possible: we can choose a segmentation with a resolution high enough to obtain a superset of the motion boundaries, in which case regions have to be successively merge based on their motion.

Another strategy, which will not be presented in this article, consists in choosing a segmentation with an intermediate resolution and to split regions if the robust motion estimation indicates that it contains more than a single type of motion. The segmentation also allows for this kind of strategy: the resolution can simply be increased within the region concerned. This allows a closed loop between segmentation and motion estimation to be established.

3 Motion Estimation

The motion information available for an individual pixel is incomplete due to the aperture problem: only the motion component normal to iso-brightness contours can be measured. Therefore, the measurements of several pixels have to be combined to obtain a complete motion vector. The regions of our segmentation provide us with an ideal support for this combination because they generally correspond to a single object in the scene.

We will first present the technique used to measure the normal motion of each pixel and then introduce the model that is used to integrate the partial motion information inside each region.

3.1 Pixel Level

We use a differential technique (without regularization) to estimate the normal motion at pixel level [4]. This yields higher precision for fine motion than standard correlation techniques and is computationally very simple.

Two images have to be prefiltered to prepare for differentiation and to increase the signal-to-noise ratio. We use a cube shaped spatio-temporal gaussian filter with a side length of 7 pixels / frames (i.e. the filtered value is calculated from the values of its neighbours in a cube, which extends 3 frames in time and 3 pixels in x and y to either side).

If we now assume that the intensity is conserved, $dI(\boldsymbol{x},t)/dt = 0$, the gradient constraint equation can be derived:

$$(\nabla I(\boldsymbol{x},t))^T \boldsymbol{v} + I_t(\boldsymbol{x},t) = 0 \ , \tag{1}$$

where $\nabla I(\boldsymbol{x},t)$ is the gradient and I_t the partial temporal derivative of the intensity. This equation gives us the motion component normal to spatial contours with constant intensity: $\boldsymbol{v}_\mathrm{n} = v_\mathrm{n}\boldsymbol{n}$, where the normal velocity and the normal direction are given by:

$$v_\mathrm{n}(\boldsymbol{x},t) = \frac{-I_t(\boldsymbol{x},t)}{\|\nabla I(\boldsymbol{x},t)\|} \qquad \text{and} \qquad \boldsymbol{n}(\boldsymbol{x},t) = \frac{\nabla I(\boldsymbol{x},t)}{\|\nabla I(\boldsymbol{x},t)\|} \ . \tag{2}$$

3.2 Region Level

In order to integrate the partial motion information of the individual pixels at region level we use a parametric model. We have decided to employ a nodal representation. A fixed number of nodes $\{\boldsymbol{x}_i\}$ is chosen depending on the motion type and complexity.

The modelling then consists in computing a "model velocity" $\boldsymbol{\phi}(\boldsymbol{x}_i)$ at each node $\boldsymbol{x}_i$, such that the interpolated velocity field based on the nodal velocities is as close as possible to the observed velocity field within the region [3].

Being velocities, the parameters of the model have a small range of variation, of the same magnitude as the motion in the sequence, which contributes to the robustness of the computations.

The interpolation technique we use is a linear technique called kriging. The velocity of each point of the region is then a linear function of the velocities at the fixed nodes:

$$\boldsymbol{v}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\}) = \sum_i \lambda_i(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x}_i) \ . \tag{3}$$

In this equation the $\boldsymbol{\phi}(\boldsymbol{x}_i)$ represent the node velocities that have to be determined, the $\lambda_i(\boldsymbol{x})$ are the corresponding weights for the interpolation given by kriging.

Using kriging as interpolation method has two important advantages. On the one hand, it is very flexible: it is possible to model the structure of the motion field by choosing an appropriate covariance model. In our case, we have chosen a covariance model yielding a spline interpolation. On the other hand, the weights only depend on the geometry, i.e. the position of the pixel and the position of the nodes to be interpolated. This allows the interpolation weights $\lambda_i(\boldsymbol{x})$ to be tabulated once and for all.

The number and the placement of the nodes determines the motion complexity that can be represented: a single node corresponds to a simple translation, three nodes with a non collinear placement yield an affine model, more than three nodes produce models with increasing complexity.

The number of motion measurements, which depends on the region's size, limits the number of model parameters that can be estimated reliably. For the integration of the motion information in each region we use models with up to four nodes (which corresponds to a maximum of eight parameters to be estimated). Figure 4 shows the chosen node placements for 3 and 4 node models.

Fig. 4. Node placement

To estimate the motion parameters of a region R we replace $\boldsymbol{v}$ in the gradient constraint (1) with our model $\boldsymbol{v}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\})$ so that the measurement in each point of R yields a constraint on the motion parameters $\boldsymbol{\phi}(\boldsymbol{x}_i)$:

$$(\nabla I(\boldsymbol{x},t))^T \boldsymbol{v}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\}) + I_t(\boldsymbol{x},t) = 0 \ . \tag{4}$$

The over-determined set of linear equations now has to be solved. The standard least-square approach consists in minimizing the following sum:

$$\sum_R ((\nabla I(\boldsymbol{x},t))^T \boldsymbol{v}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\}) + I_t(\boldsymbol{x},t))^2 \ , \tag{5}$$

which with (2) can also be written:

$$\sum_R \|\nabla I(\boldsymbol{x},t)\|^2 (\boldsymbol{n}\boldsymbol{v}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\}) - v_\mathrm{n})^2 \ , \tag{6}$$

where v_n is the normal velocity and $\boldsymbol{n}$ the normal direction.

The second expression can be interpreted as a weighted over-determined set of equations. In each point of the region the equation

$$\boldsymbol{n}\boldsymbol{v}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\}) - v_\mathrm{n} = 0 \tag{7}$$

gives the normal velocity $\boldsymbol{v}_{\mathrm{n}} = v_{\mathrm{n}}\boldsymbol{n}$. Each of these equations is weighted with $\|\nabla I(\boldsymbol{x}, t)\|^2$. The weight controls the influence of each measurement in the parameter estimation.

In general it makes sense to give higher weight to measurements from areas with high gradient: noise is less likely to corrupt those measurements (i.e. they have a higher signal-to-noise ratio).

This, however, does not hold true at motion boundaries, as will be shown in the following section.

3.3 Motion Estimation at Object Boundaries with Occlusion

We have seen that the aperture problem restricts local motion measurement to only the normal component. But even to estimate this normal component, estimation techniques have to take into account a small neighbourhood. Correlation techniques, for example, require a minimum window size, differential techniques combine information through prefiltering.

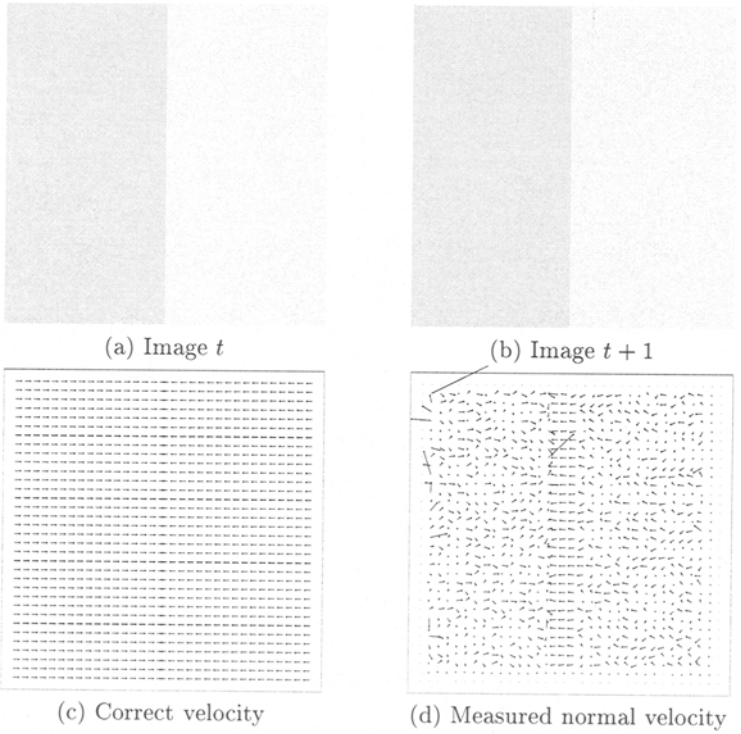


Fig. 5. Occlusion

The effect that this has on motion estimation around object boundaries will be shown in an example. Figure 5 shows a synthetic sequence (size: 100 $\times$ 100)

with occlusion: two surfaces move towards each other with the surface on the right occluding the surface on the left. The two surfaces have random texture with grey values drawn uniformly from the intervals $[200, 210]$ and $[220, 230]$. Their velocities are $(1, 0)$ and $(-1, 0)$ as shown in Fig. 5(c) (the velocity field is subsampled by a factor 3 and scaled by a factor 2). In order to compare the measured normal velocity with the known correct velocity, we evaluate the following error:

$$\text{error} = |\boldsymbol{n}\boldsymbol{v}_\text{c} - v_\text{n}| \ , \tag{8}$$

where $\boldsymbol{v}_\text{c}$ represents the known correct velocity. This is simply the difference between the measured normal velocity v_n and the projection of the correct velocity $\boldsymbol{v}_\text{c}$ onto the normal direction.

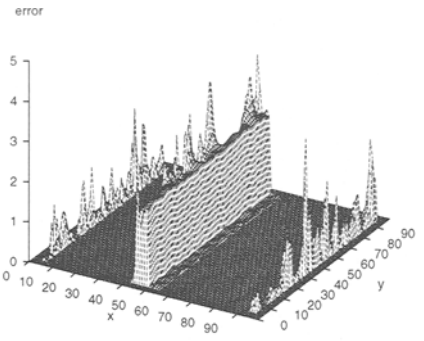
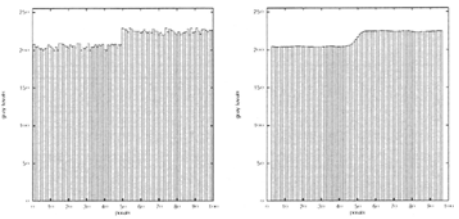


Fig. 6. Error at occlusion



(a) Before smoothing (b) After smoothing

Fig. 7. Image profile at $y = 50$

Figure 6 shows the situation around the motion boundary at $x = 50$: the error is negligible on the side of the occluding region but significant on the side of the occluded region. The error on the side of the occluded object is about 2 in a narrow band next to the contour: this indicates that the motion measured corresponds to the occluding object. This error is due to the small neighbourhood that contributes to the motion information in each pixel. Figure 7 shows how filtering affects the motion measurement: the discontinuity is smoothed and spread into the occluded region. The motion measured in this area therefore corresponds to the occluding object.

The width of the area with erroneous measurements depends on the size of the neighbourhood that contributes to the motion measurement and the relative motion of the two regions. We can observe this kind of error even if the grey level difference between the regions is very small as long as the relative motion of the regions has a non-zero component in the direction normal to the motion boundary.

The motion parameters for a simple translation estimated through minimizing (6) are $\{(-0.942, 0.004), (-0.994, 0.003)\}$. As expected, the parameters calculated for the occluded region do not reflect the correct motion (for this example they are even almost identical with the occluding region). The main reason for the bad performance is that a certain number of measurements contributing to the motion estimation are erroneous. The problem is then aggravated by the weighting with the square gradient: the erroneous measurements receive higher weight since they come from an area with high gradient. The parameters we obtain without the weighting are $\{(0.413, 0.012), (-1.002, 0.002)\}$.

In the next section we show how robust estimation techniques can help to overcome this problem.

3.4 Robust Regression

In the previous section we saw that the parameter estimation has to be able to cope with erroneous motion measurements in order to perform correctly at motion boundaries. Those motion boundaries are not the only source for erroneous measurements, all kinds of noise can corrupt the motion information.

In statistics all these erroneous measurements are known as *outliers*. Robust estimation techniques allow outliers to be detected and eliminate (or limit) their influence on the estimation: they yield the parameters that best fit the majority of the measurements [5, 8].

We have concentrated on a class of techniques called *M-estimators* that can be easily implemented as iterative reweighted least-square estimation.

First we simplify our notation. Then we show why the least-square approach we used above lacks in robustness and how M-estimators cope with outliers.

In the following we will refer to

$$|\boldsymbol{nv}(\boldsymbol{x}, \{\boldsymbol{\phi}(\boldsymbol{x}_i)\}) - v_{\mathrm{n}}| \tag{9}$$

as the absolute residuals which will be noted as $r(\boldsymbol{x})$ or simply as r.

Instead of noting the motion parameters as vectors (the node velocities $\boldsymbol{\phi}(\boldsymbol{x}_i)$) we replace them by a set of scalar parameters p_j with $j = 1, \ldots, m$, where m is twice the node number.

If we now abandon the gradient weighting, for the reasons shown above, (6) can now be written as

$$\min \sum_R r^2(\boldsymbol{x}) \ . \tag{10}$$

We see that each residual contributes with its square to the sum we have to minimize. This explains why even a single erroneous measurement with a large error can wreak havoc on the estimation: due to the squaring of the residuals the influence on the sum is so big that the parameters get pulled away from the correct solution during minimization.

M-estimator minimize the following sum:

$$\min \sum_R \rho(r/\hat{\sigma}) \ . \tag{11}$$

In this equation ρ is the function that replaces the square and $\hat{\sigma}$ is a robust estimate for the standard deviation of the residuals which serves as a scale parameter. The robust estimate $\hat{\sigma}$ is given by

$$\hat{\sigma} = 1.4826[1 + 5/(n - m)] \text{ median}(r) \ , \tag{12}$$

where m is the number of parameters to be estimated and n the number of measurements in the region.

In order to be able to cope with outliers, the function ρ has to be less increasing than square. The literature proposes a multitude of functions from which we have retained the functions *Fair* and *Geman-McClure* whose formulas are given in Table 1 and which are depicted in Fig. 8.

type	$\rho(x)$	$\psi(x)$	$w(x)$
L_2	$\frac{x^2}{2}$	x	1
Fair	$c^2\left[\frac{\lvert x\rvert}{c} - \log(1 + \frac{\lvert x\rvert}{c})\right]$	$\frac{x}{1+\lvert x\rvert/c}$	$\frac{1}{1+\lvert x\rvert/c}$
Geman-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{(1+x^2)^2}$	$\frac{1}{(1+x^2)^2}$

Table 1. Least-square and used M-Estimators

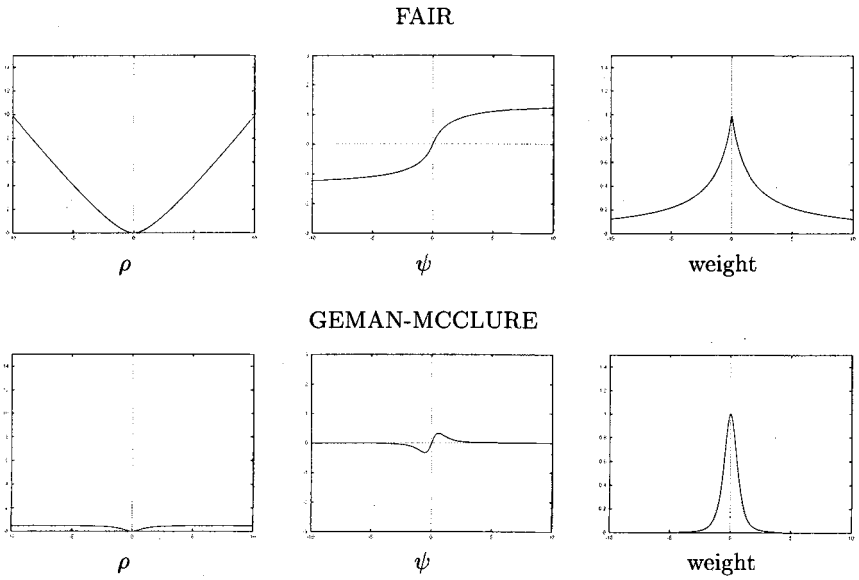


Fig. 8. Plots of the used M-Estimators

We will now show how the minimization of (11) can be solved as an iterated reweighted least-square problem.

The minimization of (11) is equivalent to the solution of the following linear system:

$$\sum_R \psi(r/\hat{\sigma}) \frac{\partial r}{\partial p_j} = 0, \; j = 1, \ldots, m \;, \tag{13}$$

where $\psi = d\rho(x)/dx$ is called the *influence function*. If we then define a *weight function*

$$\omega(x) = \frac{\psi(x)}{x} \;, \tag{14}$$

(13) can be written as

$$\sum_R \omega(r/\hat{\sigma}) r \frac{\partial r}{\partial p_j} = 0, \; j = 1, \ldots, m \;. \tag{15}$$

The previous equation is now equivalent to the following weighted least-square problem

$$\min \sum_R \omega(r/\hat{\sigma}) r^2 \;, \tag{16}$$

where $\omega(r/\hat{\sigma})$ represents the weight for each of the residuals.

Figure 9 illustrates the iterative nature of the solution.

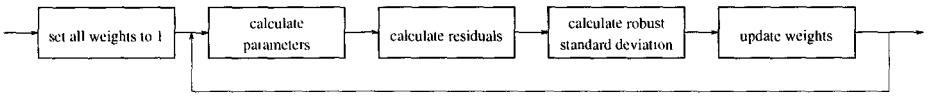


Fig. 9. Robust regression scheme

Without any prior knowledge about the reliability of the individual measurements or the model parameters, we start by setting all the weights to 1. We then calculate the parameters p_j and the corresponding residuals. The robust estimate for the standard deviation is evaluated and serves to scale the residuals for the weight computation.

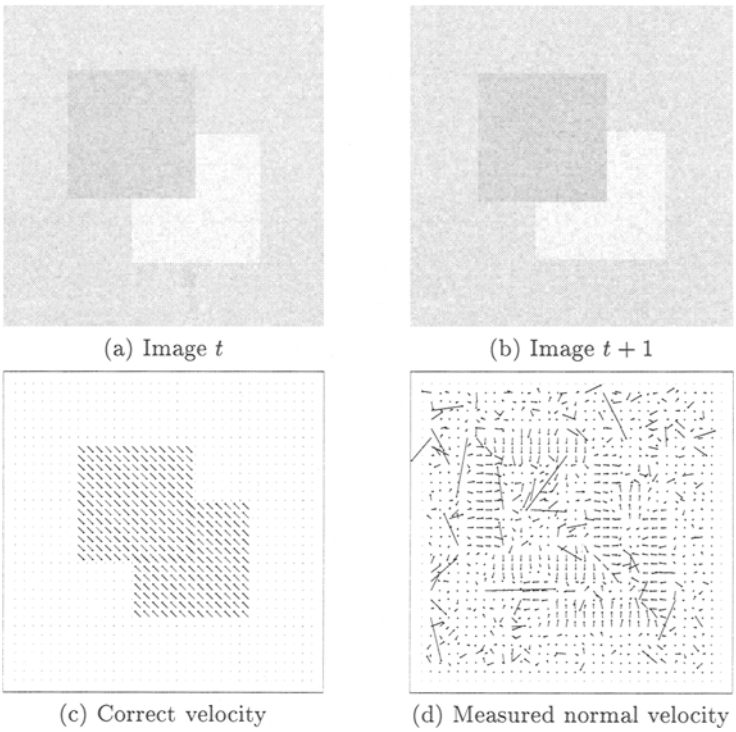
In Fig. 8 we can see that large residuals are assigned low weights for the next iteration and therefore their influence is reduced.

The main difference between the two functions ρ we use is that the Geman-McClure function can completely exclude residuals from the modelling by assigning zero weights, whereas the Fair function always yields non-zero weights. The Geman-McClure function is more severe but does not always guarantee good convergence: we therefore use the Fair function at the beginning of the iteration.

The iteration stops when $\hat{\sigma}$ no longer decreases or when a fixed number of iteration is reached.

If we apply the robust approach to our occlusion problem we, in fact, obtain the correct motion parameters $\{(1,0),(-1,0)\}$.

Let us now look at a more challenging problem with multiple occlusions and added random noise.

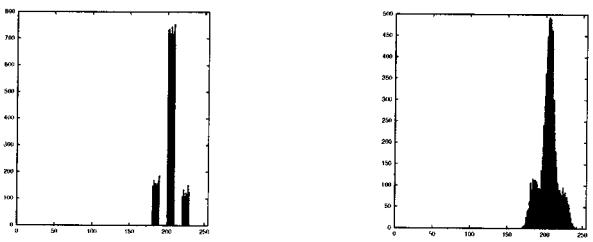


(a) Image t (b) Image $t+1$

(c) Correct velocity (d) Measured normal velocity

Fig. 10. Multiple occlusions

Figure 10 (a) and (b) show two diagonally translating squares in front of a stationary background. Their velocities are $(1,-1)$ for square 1 (top left) and $(-1,1)$ square 2 (bottom right) as shown in (c). As for the previous synthetic example, the different textures are obtained by drawing grey values uniformly from the intervals $[180,190]$, $[200,210]$ and $[220,230]$. To all the pixels of the resulting sequence Gaussian noise with $\sigma = 5$ is added independently (Fig. 11 shows the histogram before and after the addition of the noise).



(a) Without noise (b) With noise

Fig. 11. Histograms

This noise introduces significant error into the normal motion measurements (c.f. Fig. 10 (d)).

The following table compares the translation calculated with the least-square (with and without gradient weighting) and the robust parameter estimation:

	Background	Square 1	Square 2
Least-square (weighted)	$(0.032, -0.062)$	$(0.273, -0.266)$	$(-0.037, 0.035)$
Least-square	$(-0.016, -0.071)$	$(0.548, -0.670)$	$(-0.369, 0.296)$
Robust	$(0.032, -0.070)$	$(0.993, -0.999)$	$(-0.940, 0.960)$

The fact that the background motion is well estimated in all three cases is due to the opposed motion of the identical squares whose influence on the occluded background cancels itself out.

Figure 12 shows the location of the measurements that have been classified as outliers (in grey) along with the contours of the objects (in black). Most of the outliers fall into areas with low gradient where the signal-to-noise ratio is very low. We can also see a high concentration of outliers where square 1 occludes square 2: as expected, the outliers are found on the side of the boundary that corresponds to the occluded object.

Fig. 12. Outliers

4 Region Merging

Now that we are able to calculate the correct motion parameters for the regions of the segmentation, we can group the regions that have similar motion. This will be done as an iterative region merging: at each iteration, all pairs of adjacent regions are candidates for merging. Instead of trying to compare the motion in the parameter space, we calculate a new set of motion parameters for each of the region pairs and evaluate the resulting modelling quality. Quality measures based on the motion compensated images (i.e. PSNR) have been tested but have proven inconsistent and time consuming. We use the mean modelling error of the motion instead:

$$\text{error}(R, \boldsymbol{p}) = \frac{1}{\text{size}(R)} \sum_{R} \omega(\boldsymbol{x})(\boldsymbol{n}\boldsymbol{v}(\boldsymbol{p}) - v_{\text{n}})^2 \ , \tag{17}$$

where $\omega(\boldsymbol{x})$ are the weights we have calculated through robust regression and $\boldsymbol{p}$ our m motion parameters.

The merging criterion can then be based on the individual errors before the merging $\{\text{error}(R_1, \boldsymbol{p}_1), \text{error}(R_2, \boldsymbol{p}_2)\}$ and the modelling errors when the joint model parameters have been used $\{\text{error}(R_1, \boldsymbol{p}_{12}), \text{error}(R_2, \boldsymbol{p}_{12})\}$.

If two regions have similar motion, the jointly calculated motion parameters yield small errors $\{\text{error}(R_1, \boldsymbol{p}_{12}), \text{error}(R_2, \boldsymbol{p}_{12})\}$ when applied to the individual regions: we therefore consider the motion of two regions as similar if the following criterion

$$C = \max\{\text{error}(R_1, \boldsymbol{p}_{12}), \text{error}(R_2, \boldsymbol{p}_{12})\} \tag{18}$$

is small.

The different steps of the merging procedure with an exhaustive evaluation of the similarity criterion C for all region couples are the following:

1. Evaluation of the similarity criterion C for all couples of adjacent regions,
2. merging of the couple with the most similar motion (smallest C),
3. updating of the criteria for all the region pairs involved in the merging (i.e. all the region couples that contained one of the two merged regions),
4. iteration from point 2.

This exhaustive approach at first seems very costly. However this is not the case: the joint modelling of two regions requires only the solution of one over-determined linear system in the least-square sense (since we keep the weights already established through robust estimation) for which efficient numerical tools exist. In particular, we may reuse parts of the calculus for the individual motion parameters to calculate the joint parameters.

The merging will be stopped when a predefined error threshold is exceeded or when the criterion rises abruptly. The functioning of the merging will be shown along with the results of the depth ordering we shall introduce in the next section.

5 Depth Ordering

As seen in Sect. 3.3, occlusion causes significant error on the side of the occluded object. This error makes it possible to deduce the depth ordering of the involved objects.

As mentioned before, measurements at locations with high gradient are less sensitive to noise and thus yield more reliable values. We therefore have to distinguish two main classes of outliers: regions with low gradient which are likely to be corrupted by noise and regions with high gradient which generally correspond to occluded objects (or if the segmentation is not fine enough they might also indicate multiple types of motion in a region).

Figure 12, which we have already seen, shows both types: spread unevenly across the image we find outliers that correspond to the first class; at the motion boundary between the two squares the outliers are due to occlusion.

A simple and elegant way to separate the two types of outliers is to make use of the gradient information: we weight the absolute residuals $r(\boldsymbol{x})$ obtained

through robust estimation with the modulus of the gradient

$$r_{\text{weighted}}(\boldsymbol{x}) = |\nabla I(\boldsymbol{x}, t)| r(\boldsymbol{x}) \ , \tag{19}$$

and reestimate $\hat{\sigma}$. We then recalculate the weights $\omega(r/\hat{\sigma})$ with the chosen weight function. The measurements which are now classified as outliers (i.e. have zero weight) correspond to true modelling errors: the occluded regions.

The location of the measurements that are classified as outliers in this way are shown in Fig. 13.

In order to quantify this information we compute the spatial outlier density in a narrow band on either side of the motion frontier. The width of the band depends linearly on the relative motion of the two regions normal to the contour and on the size of the prefilter used. For our test we have used a width of three pixels.

Figure 14(a) shows these bands for the occluding square example.

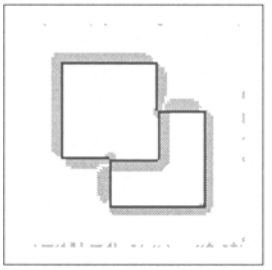


Fig. 13. Outliers (weighted)

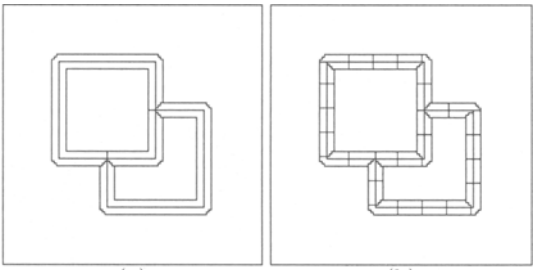


Fig. 14. Bands used in the depth evaluation

We now have to establish an ordering based on these two densities. If we define two thresholds t_{low} and t_{high} we can distinguish between situations with and without a clear depth ordering. A clear ordering exists when one density is below t_{low} and the other above t_{high}. In all other cases, we cannot make any statement about the ordering. The low threshold allows for a certain number of false outliers and the high threshold indicates the minimum number of outliers for a region to be considered occluded.

In this initial form, the approach only works for simple cases. Let us consider the situation where a narrow rectangle translates in the direction of its longer sides in front of a stationary background. Error due to occlusion can only be observed at the short sides: the outlier density in the background therefore will be very small and normally does not exceed t_{high}. This is why we partition the bands into short strips (c.f. Fig. 14(b)) and use a kind of "voting" mechanism: only the pairs of strips with a clear ordering contribute to the depth detection. With this approach, a correct depth ordering becomes possible for the previous example and for most natural scenes.

In our examples, we have used a length of 20 pixels for the partitioning of the bands and we set the threshold to $t_{\text{low}} = 0.2$ and $t_{\text{high}} = 0.8$.

This relative ordering is represented in the form of a directed graph: nodes correspond to the regions and the edges indicate relative depth. We can now

perform (if there are no cycles) a topological sort on the graph: as a result, we obtain an image in which low grey values correspond to objects close to the observer and high grey values to more distant ones.

Figure 15 shows the different depths for the translating square sequence (with white indicating the most distant object).

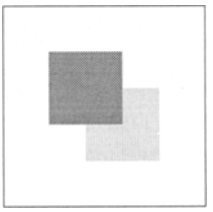


Fig. 15. Depth

Let us now see how the algorithm performs on natural scenes. The results for the image sequences "Foreman" and "Claire" are show in Fig. 16 and 17 (the velocity fields are subsampled by a factor 5 and scaled by a factor 4). The image size, the number of the frame treated in the sequence and the number of regions used in the segmentation are given in the following table:

name	size	frame no.	region no.
"Foreman"	QCIF (176×144)	50	75
"Claire"	QCIF (176×144)	15	20

For these examples a model with three nodes which is capable of representing affine motion has been used.

The normal velocity for "Foreman" in Fig. 16(b) shows two different types of motion: the upper body moves to the left and the background moves up to the right (due to the camera motion down to the left). As we have already seen, the segmentation with 75 regions contains the major motion boundaries. We can also see that the region merging is then correctly performed. Note however that there is a small region on the right that has merged with the person's shoulder although it belongs to the background. This is due the fact that the region is relatively narrow: the majority of its motion measurements yield the motion of the occluding object, in which case the robust estimation produces the foreground motion. This problem can easily be resolved by imposing a minimum region size or a morphological constraint on the segmentation's regions. The depth ordering then yields the correct depths: the person shown in grey is situated in front of the background in white.

The motion in the sequence "Claire" (Fig. 17(b)) is quite small. We can see that the head moves downwards and that the upper body and the background are practically still. Also note that some of the motion measurements in the background (mostly at the top and on the right), due to some form of interference, indicate large motion. Due to the relative simplicity of the grey-level image, a region number of 20 is sufficient for the segmentation. The correct result of the merging shows that the robust technique has coped well with the

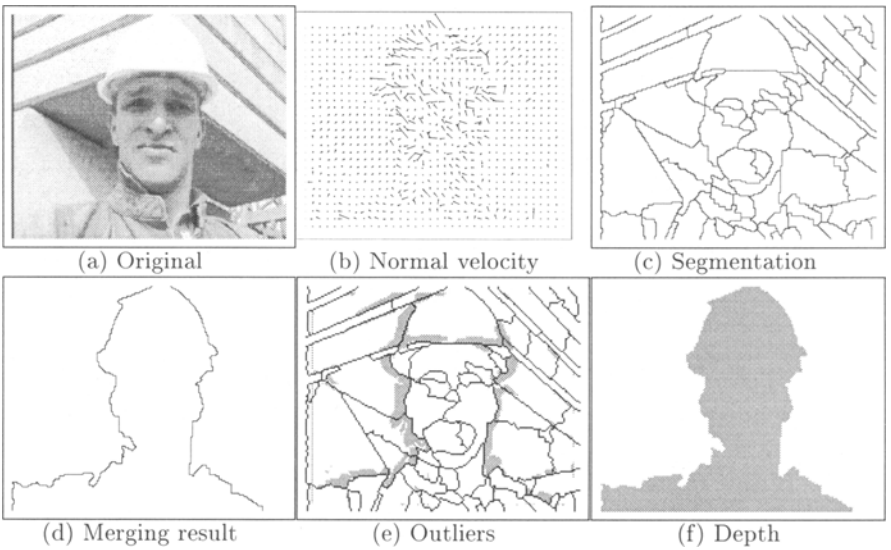


Fig. 16. "Foreman"

erroneous measurements in the background. The calculated depth correctly reflects the structure of the scene, which demonstrates that the depth ordering also performs well for small motion.

6 Conclusion

We have shown how morphological segmentation, combined with robust parameter estimation techniques, can be used to segment the motion of a scene with multiple occluding objects. We have then presented a new technique that performs depth ordering of the resulting image partition: the modelling error at the motion boundaries is used to indicate the occlusion relationship.

The main advantage of this approach lies in the fact that the motion has to be analyzed between only two frames to perform the depth ordering, unlike existing techniques [2, 6, 7] which require three frames.

This becomes possible through the combination of the morphological segmentation, which provides precise contour placement, and the robust estimation, which indicates modelling errors due to occlusion.

The effect that this has on performance is twofold. It allows to deduce depth ordering even if the motion is very small and it provides high robustness.

The use of a closed loop containing segmentation and motion estimation, as mentioned above, is a step towards more flexibility. A step towards more stability then consists of the integration of motion and depth information across several image frames. The information at time t can be used to initialize segmentation, robust motion estimation and depth ordering at time $t+1$. All this information can then be accumulated over multiple frames which will allow a scene to be segmented correctly into objects even if, temporarily, no motion is present.

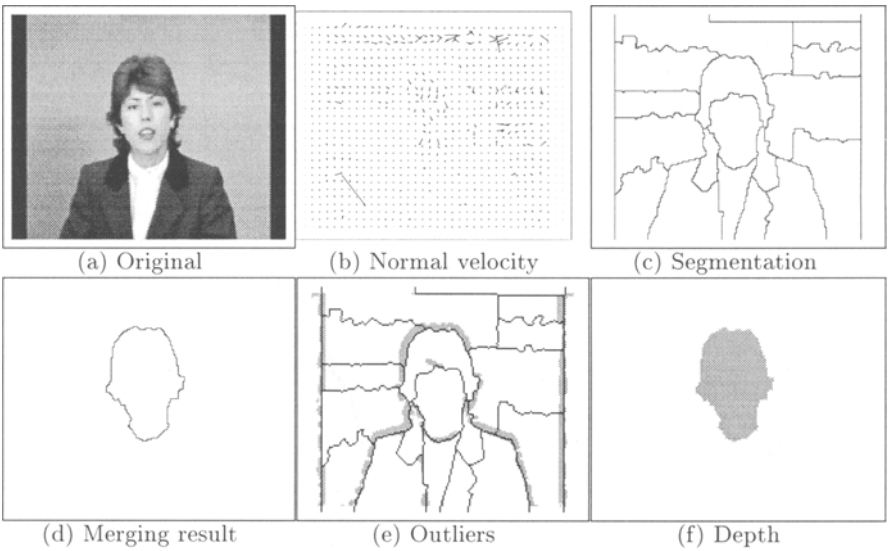


Fig. 17. "Claire"

Acknowledgments

This work has been financed by CNET France Telecom.

References

1. J. Cichosz and F. Meyer. Morphological multiscale image segmentation. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'97)*, pages 161–166, Louvain-la-Neuve (Belgium), June 1997.
2. Trevor Darrell and David Fleet. Second-order method for occlusion relationships in motion layers. Technical Report 314, MIT Media Lab Vismod, 1995.
3. E. Decencière Ferrandière, C. de Fouquet, and F. Meyer. Applications of kriging to image sequence coding. *Accepted for publication in Signal Processing: Image Communication*, 1997.
4. B. K. P Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
5. Peter Meer, Doron Mintz, Dong Yoon Kim, and Azriel Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, April 1991.
6. K. M. Mutch and W. B. Thompson. Analysis of accretion and deletion at boundaries in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:133–138, 1985.
7. W. B. Thompson, K. M. Mutch, and V. A. Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:374–383, 1985.
8. Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report 2676, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis Cedex, France, October 1995.