Computation of Discontinuous Optical Flow by Domain Decomposition and Shape Optimization*

C. Schnörr

Fraunhofer – Institut für Informations– und Datenverarbeitung (IITB)
Fraunhoferstrasse 1
D – 7500 Karlsruhe 1
Federal Republic of Germany

We introduce an approach for the identification and tracking of image regions in monocular image sequences.

The assumptions are that the egomotion is known and that an representation of the environment has been computed in order to predict roughly the expected motion field with respect to the movement of the sensor relative to the environment.

Image regions, whose motions do not match with this motion field, are assumed to belong to independently moving objects in the scene. Having detected roughly the location and movement of such a region in the image plane (by Optical Flow or any other approach), we decompose the image plane in two disjoint domains and alter their shape such that the velocity fields which are computed for both domains agree in an optimal way with our interpretation of the scene.

By this method we reconstruct a velocity field for the whole domain of the image plane, which is discontinuous along a closed contour. This contour coincides in an optimal, just described way with the motion boundaries of the object. As a result, a segmentation of the velocity field a posteriori becomes superfluous.

Let $\mathcal{H}$ denote the Hilbert space $H^1(\Omega) \times H^1(\Omega)$. ($H^1(\Omega)$ denotes as usual the Sobolev space $H^1(\Omega) \equiv W_2^1(\Omega)$; cf. [8]). The approach of Horn and Schunck [2] for the reconstruction of a velocity field $u = (u1, u2)^T$ for a domain $\Omega \subset \mathrm{R}^2$ can be formulated as follows (cf. [5]):

find $u \in \mathcal{H}$ such that

$$J_{H/S}(u) = \inf J_{H/S}(v) \quad , \quad v \in \mathcal{H}$$

where the functional $J_{H/S} : \mathcal{H} \rightarrow \mathrm{R}$ (H/S from Horn/Schunck) is defined by

$$J_{H/S}(v) = \frac{1}{2} a(v, v) - f(v) + const \; . \qquad (1)$$

Here $a(\cdot,\cdot) : \mathcal{H} \times \mathcal{H} \rightarrow \mathrm{R}$ is the symmetric, bounded bilinear form

$$a(u, v) = 2 \int_\Omega \{ (\nabla g \cdot u)(\nabla g \cdot v) + \lambda^2 (\nabla u1 \cdot \nabla v1 + \nabla u2 \cdot \nabla v2) \} \, dx \; , \qquad (2)$$

$f(\cdot) : \mathcal{H} \rightarrow \mathrm{R}$ is the continuous linear form

$$f(v) = -2 \int_\Omega \{ g_t g_x v1 + g_t g_y v2 \} \, dx \; , \qquad (3)$$

and the constant term is given by

$$const = \int_\Omega g_t^2 dx \; . \qquad (4)$$

It can be shown (see [5]) that, under weak conditions on the greyvalue gradient, the functional (1) with (2),(3) and (4) is strictly convex and that the unique solution of this minimization problem can be computed by solving the variational equality

$$a(u, v) = f(v) \; , \qquad \forall v \in \mathcal{H} \qquad (5)$$

By discretisation with Finite Elements, this problem reduces to the solution of a linear system with a symmetric, sparse and positive definite matrix.

Explicitely, (1) reads as

$$J_{H/S}(v) = \int_\Omega \{ (\nabla g \cdot v + g_t)^2 + \lambda^2 (|\nabla v1|^2 + |\nabla v2|^2) \} \, dx \qquad (6)$$

Fig. 1 shows a velocity field, which was computed according to this approach. Assuming that the camera is stationary, one would hypothesise here the presence of a moving object in the scene and would try to determine the corresponding image region, i.e. the motion boundary. The vector field, which corresponds to the projected motion on the image plane, is in general discontinuous on motion boundaries. In contrast to this, it is well known that the variation of a velocity field, which is reconstructed by the minimization of (1), is bounded and that, consequently, the detection of motion boundaries a posteriori, for example with a threshold, is difficult.

Our approach is to introduce motion boundaries explicitely by decomposing the domain Ω and to move them in the image plane by minimizing a suitable criterion. To this end we assume, as stated above, that the motion within the image plane can be roughly predicted

*This work has been supported by the PROMETHEUS programme of the Bundesministerium für Forschung und Technologie, Bonn and the following companies: BMW AG, Daimler–Benz AG, MAN AG, Dr.h.c. F.Porsche AG, and Volkswagen AG.

BMVC 1990 doi:10.5244/C.4.21

for each domain Ω_k in terms of a model vectorfield u_{km}, and minimize

$$J(\Omega_k) = \int_{\Omega_k} |u_k - u_{km}|^2 dx \,, \qquad (7)$$

where u_k is the solution of eqn. (5) (w.r.t. Ω_k), and Ω_k is the "variable" which has to be determined.

Note that the assumption that a vectorfield u_{km} is available is primarily related to the situation, where we want to identify an image region. Here we have in view the embedding of our approach into a vision system, which uses several different modules in order to estimate 3D–parameters. Such a system should be possible to predict a motion field, which roughly corresponds to the movement of the sensor relative to the environment. Concerning the tracking of an image region, the vectorfield corresponding to the last image should provide enough information to compute u_{km} (see below).

Outline of the paper

The remainder of our contribution is organized as follows: In the next section, we sketch our algorithm for the identification of an image region from Optical Flow (for lack of space, we omit the derivation, which can be found in [6] (as well as a discussion of related literature)). In the third section, we show experimental results, which demonstrate the feasibility as well as some specific properties of our approach. Finally, we indicate some directions of further research.

A descend procedure for the identification of image regions from Optical Flow

We sketch now briefly our procedure, which generates iteratively a sequence $\{\Omega^n\}$ of domains $\Omega^k \subset \mathrm{R}^2$, such that

$$J(\Omega^{k+1}) \leq J(\Omega^k) \qquad (8)$$

Details may be found in [6]. Choosing for arbitrary k the notation

$$\Omega := \Omega^k \,,\ \Omega' := \Omega^{k+1} \,, \quad u := u_k \,,\ u_m := u_{km} \,,$$

we consider Ω' as a perturbation of the domain Ω by a vectorfield $V \in \mathcal{V}$

$$\Omega' = (I + V)(\Omega) = \{x + V(x) \,|\, x \in \Omega\} \,,$$

where $I = \mathrm{id}_{\mathrm{R}^2}$ and V is an element of a space $\mathcal{V}$ of smooth vectorfields which are defined on R^2. V should be "sufficiently small" in order to ensure that the mapping $I + V$ is a diffeomorphism of R^2. We use results of Murat and Simon [4, 7] about the (formal) differentiability of mappings from a family of domains

$$\mathcal{D} := \{\Omega' \,|\, \Omega' = (I + V)(\Omega)\}$$

into spaces $W_p^k(\Omega)$ of smooth functions and R^1, respectively, and exploit a standard trick from the theory about the control of distributed parameter systems (introduction of an "adjoint state" p_k, see below; cf. [3], for example) in order to calculate the derivative of the functional (7) at the point $V = 0$ in the direction of a field $\tilde{V}$. Finally, we get an estimate of the sign of the normal component on $\partial\Omega$

$$\tilde{V}_n := n \cdot \tilde{V}$$

of vectorfields $\tilde{V}$, for which we have (8), by the formula

$$\begin{aligned} \tilde{V}_n \;=\; & -\big[|u - u_m|^2 - 2(\nabla g \cdot u)(\nabla g \cdot p) \\ & -2\lambda^2(\nabla u1 \cdot \nabla p1 + \nabla u2 \cdot \nabla p2) \\ & -2g_t(\nabla g \cdot p)\big] \quad \text{on} \quad \partial\Omega \end{aligned}$$

where, see Fig. 2, n denotes the outer normal of Ω and the vectorfield $p = (p1, p2)^T$ is the solution of

$$a(p, v) = 2\int_{\Omega} (u - u_m) \cdot v dx \qquad \forall v \in \mathcal{H} \,. \qquad (9)$$

For the situation shown in Fig. 3 we define analogously

$$J = \int_{\Omega_i} |u_i - u_{im}|^2 dx + \int_{\Omega_o} |u_o - u_{om}|^2 dx \qquad (10)$$

with "i" for inside and "o" for outside. Assuming that $\tilde{V} = 0$ on $\partial\Omega_o \backslash (\bar{\Omega}_o \cap \bar{\Omega}_i)$ we get

$$\begin{aligned} \tilde{V}_n \;=\; & -\big[|u_i - u_{im}|^2 - 2(\nabla g \cdot u_i)(\nabla g \cdot p_i) \\ & -2\lambda^2(\nabla u1_i \cdot \nabla p1_i + \nabla u2_i \cdot \nabla p2_i) \\ & -2g_t(\nabla g \cdot p_i) \\ & -|u_o - u_{om}|^2 + 2(\nabla g \cdot u_o)(\nabla g \cdot p_o) \\ & +2\lambda^2(\nabla u1_o \cdot \nabla p1_o + \nabla u2_o \cdot \nabla p2_o) \\ & +2g_t(\nabla g \cdot p_o)\big] \\ & \text{on} \quad \Gamma = \bar{\Omega}_o \cap \bar{\Omega}_i \,. \end{aligned}$$

Therefore, one iteration of the descend procedure comprises the following steps:

1. Compute u_i, u_o according to (5)
2. Compute p_i, p_o according to (9)
3. Move the boundary Γ (w.r.t. Ω_i) to the outside (to the inside), if $\tilde{V}_n > 0$ ($\tilde{V}_n < 0$).

An extension of the approach to the case of several objects is obvious.

Experimental results

To keep the approach efficient, we used a uniform mesh for the discretisation of the problem and the movement of boundary markings (i.e. the "stepsize" is constant). We moved a boundary marking to the inside or to the outside if the same direction was computed for both neighbours in order to avoid very irregular shapes of the domains (and, in turn, poor triangulations) as well as oscillations in the neighbourhood of the minimum.

We generated an image sequence by moving a domain with a circle–like boundary (Fig. 4) with the constant velocity $(1, 0)^T$ pixel/frame in the image. The domain as well as the image were masked with a constant grey-value and additive white noise. As model vectorfields u_{im}, u_{om} we used the exact vectorfields.

Fig. 7 shows from the left to the right and from the top to the bottom 16 iterations of the moving boundary. The boundary was marked with black before an iteration step. After the step the boundary was drawn with white into the same image. At the beginning we started with a small circle in the detected region according to Fig. 1 (Fig. 7, first image on the top).

Fig. 7 and the values of the functional (10) for each step (Fig. 5) show that the iteration procedure runs into a (local) minimum. Fig. 6 shows the discontinuous vectorfield after the 16th step. Fig. 8 shows the result for a different starting point (the final result differs from the previous, because the update–rule for the boundary markings (see above) makes the boundary to some extent rigid).

Fig. 9 shows a real image. The camera is moving towards the crossing and a car is coming from the left. We used the part of the image shown in Fig. 11. In order to get the model vectorfield u_{om} we computed a velocity field for the lower half of the image, and estimated the parameters, which belong to a flow field caused by a moving plane, by a least-squares procedure. These parameters then were used to define u_{om} for the whole part of the image. At the beginning of the iteration procedure we decomposed the image within the image region of the car, and used the mean value of the corresponding flow field to define constant velocity fields u_{im} for the domains Ω_i^k in the subsequent iteration steps.

Fig. 10 and Fig. 11 show the identified domain. We also used a different value of the regularization parameter λ^2 (25 instead of 5) for the computation of the flow fields. Fig. 12 indicates that this has only little influence on the result.

From this result we conclude the following:

- The approach does not need prominent greyvalue–structures (as corners or edges) but evaluates the coherent motion of image regions. It is only assumed, that the partial derivatives of the greyvalue function can be (roughly) estimated ("short range motion").

- It is necessary to know (again roughly) the motion of an image region, which has to be identified. The resulting velocity field, however, can be used to identify the corresponding region in the next image, i.e., to track the region.

- The approach comes to a decision with respect to the locality of the motion boundary of an object (see Fig. 7 and Fig. 8). This makes a segmentation a posteriori superfluously.

- In general, greyvalue edges need not correspond to motion boundaries. Using edges one first has to select the correct ones (namely those that do not correspond to surface markings) and then one has to link them to a closed contour, whereas the resulting boundary of our approach is always closed.

- Because the approach avoids the usual smoothing over motion boundaries of the approach of Horn and Schunck [2] the resulting velocity field is less "distorted" (compare Fig. 1 with Fig. 6). This should make it possible to track an identified image region. For the example shown above one could, for instance, use the value

$$v = \frac{1}{|\Omega_i|} \int_{\Omega_i} u_i dx$$

to define a constant model vectorfield u_m (which is always assumed to be defined on Ω^k for each iteration step k) for the identification of the region in the next frame and to rigidly translate the boundary in order to get a good starting point for the next iteration.

- Each iteration step moves the whole boundary. For this, a vectorfield p_k has to be computed for each domain. Fortunately, the compilation of the corresponding matrix is not necessary, because this matrix is identical to the matrix which corresponds to the vectorfield u_k (compare eqn. (5) with eqn. (9)).

Further work

It might be worthwhile to apply the descend procedure to the Finite–Element equations (i.e. after the discretisation) in order to get not only the qualitative information "move the boundary at this point to the inside/outside" but also quantitative information for the control of the "stepsize" and for the development of termination criteria.

It is necessary to design a multi–grid procedure (cf. [1]) to solve the systems (5) und (9) in an optimal way. Extensive numerical investigations are then in order to investigate the sensitivity of the approach as well as effects which might arise from the presence of more than one object.

Note that any strictly convex quadratic functional for the reconstruction of Optical Flow instead of (1) fits into our approach. Similarly, alternatives with respect to (7) are possible. If one knows a priori, for example, that the velocity field, which has to be reconstructed, is nearly piecewise constant, then one could try

$$J(\Omega_k) = \int_{\Omega_k} \{|\nabla u1|^2 + |\nabla u2|^2\} dx$$

with the obvious advantage, that one does not need to know the value of the vectorfields in each domain.

Thanks to Kostas Daniilidis, Hans–Hellmut Nagel, Joachim Rieger and Karl Rohr for their comments on this paper.

References

[1] W.Hackbusch, "Multi–Grid Methods and Applications", Springer–Verlag, Berlin Heidelberg New York Tokyo, 1985

[2] B.K.P.Horn, B.G.Schunck, "Determining optical flow", *Artif. Intell.* 17, 1981, 185–203

[3] J.-L.Lions, "Optimal control of deterministic Distributed Parameter Systems", *Distributed Parameter Systems*, W.H.Ray, D.G.Lainiotis (eds.), Marcel Dekker, Inc., New York Basel 1978, 9–46

[4] F.Murat, J.Simon, "Etude de problemes d'optimal design", *Optimization Techniques, Proc. 7th IFIP Conference*, Nice, Sept. 8–12, 1975, J.Cea (ed.), Lect. Not. in Comp. Science 41, Springer–Verlag Berlin Heidelberg New York 1976

[5] C. Schnörr, "Determining Optical Flow for Irregular Domains by Minimizing Quadratic Functionals of a Certain Class", *Int. J. of Comp. Vision*, accepted for publication

[6] C. Schnörr, "Computation of Discontinuous Optical Flow by Domain Decomposition and Shape Optimization", in preparation

[7] J.Simon, "Differentiation with respect to the domain in boundary value problems", *Numer. Funct. Anal. and Optimiz.*, 2(7&8), 1980, 649–687

[8] W.I.Smirnov, "Lehrgang der höheren Mathematik", Teil V, VEB Deutscher Verlag der Wissenschaften, Berlin 1976

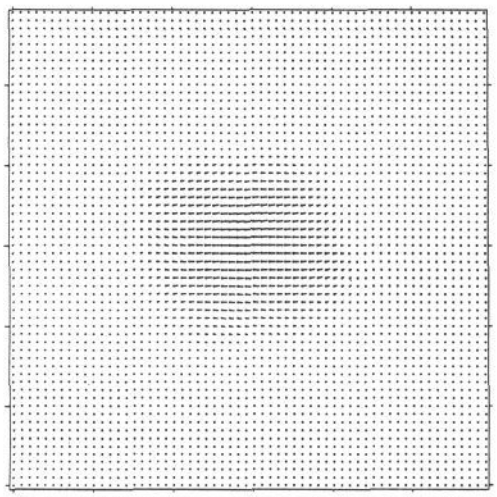

Figure 1: velocity field computed according to the approach of Horn and Schunck

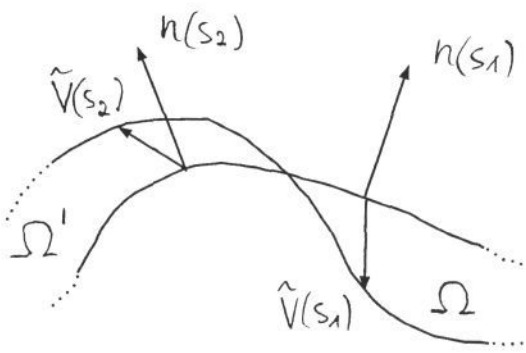

Figure 2: deformation of the domain Ω

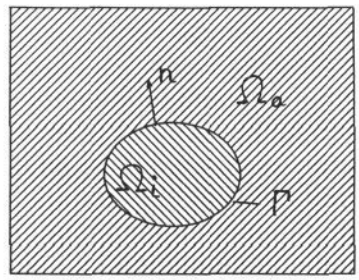

Figure 3: decomposition of the image plane

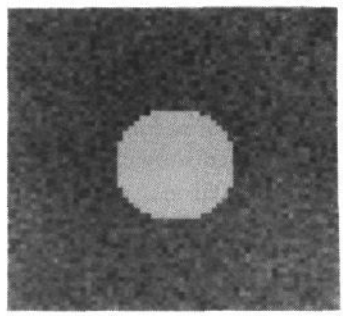

Figure 4: shape of the moving image region (white)

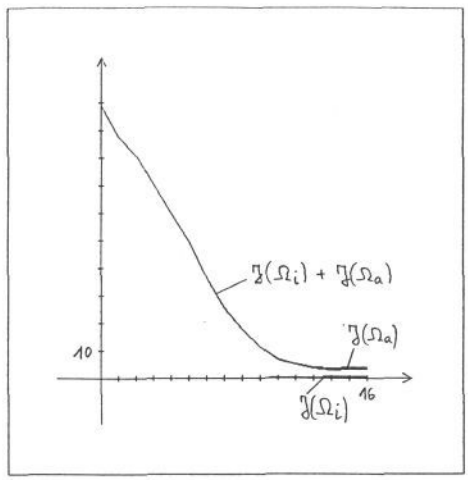

Figure 5: values of the functional (10) for the 16 iteration steps

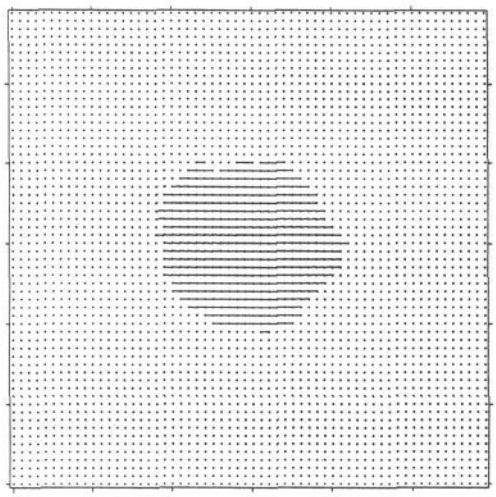

Figure 6: the reconstructed discontinuous velocity field

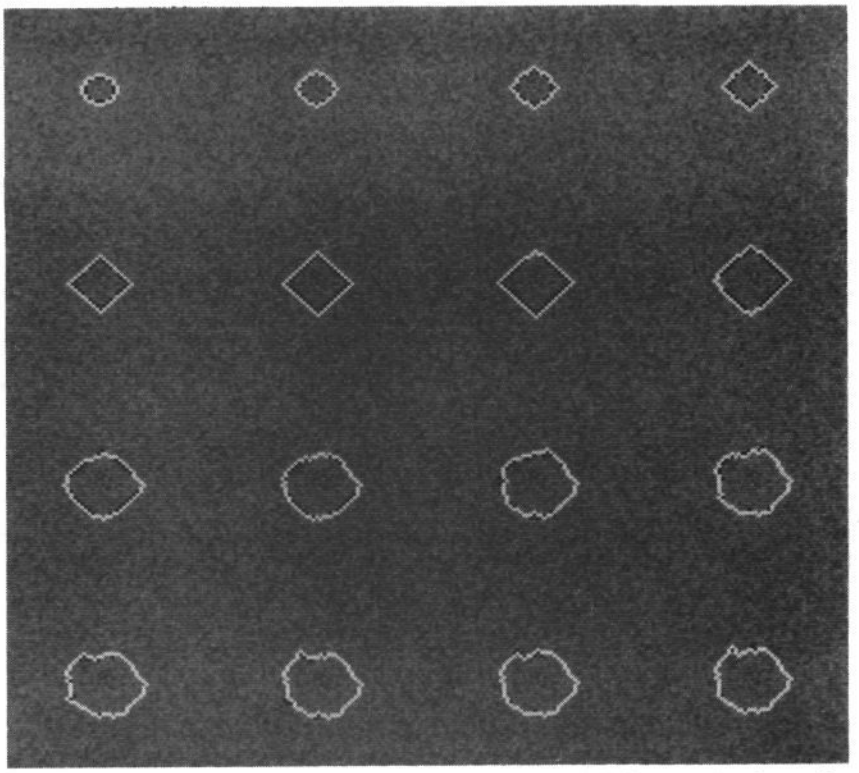

Figure 7: identification of the domain: 16 iterations of the moving boundary

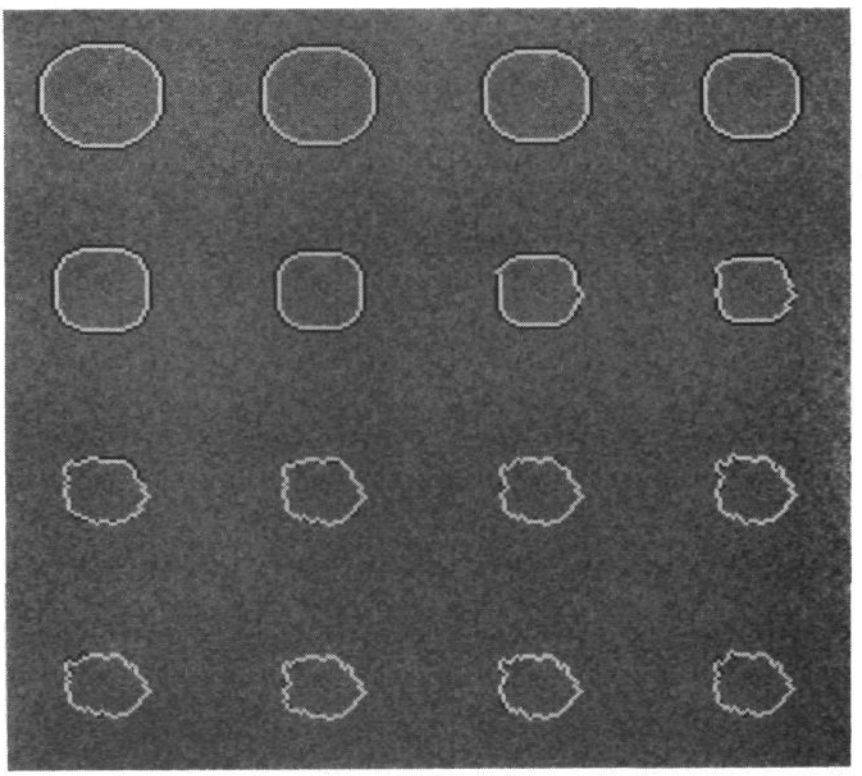

Figure 8: same as above with a different starting point

Figure 9: a real image sequence taken with a moving camera

Figure 10: the identified domain corresponding to a car coming from the left

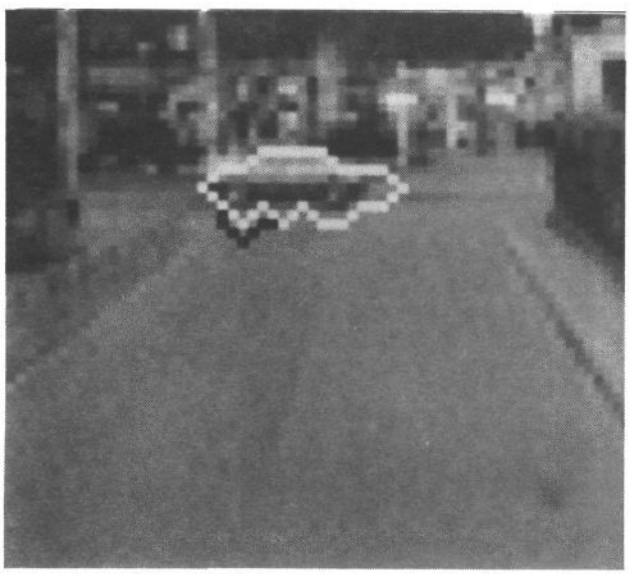

Figure 11: detail of Figure 10

Figure 12: result for a different value of the regularization parameter (see text)