Fast Obstacle Detection for Urban Traffic Situations

U. Franke, S. Heinrich
DaimlerChrylser AG
D-70546 Stuttgart, Germany
HPC: T728
{uwe.franke, stefan.heinrich}@DaimlerChrysler.com

ABSTRACT

The early recognition of potentially harmful traffic situations is an important goal of vision based driver assistance systems. Pedestrians, in particular children, are highly endangered in inner city traffic. Within the DaimlerChrysler UTA (Urban Traffic Assistance) project, we are using stereo vision and motion analysis in order to manage those situations. The flow/depth constraint combines both methods in an elegant way and leads to a robust and powerful detection scheme.

A ball bouncing on the road often implies a child crossing the street. Since balls appear very small in the images of our cameras and can move considerably fast, a special algorithm has been developed to achieve maximum recognition reliability.

I. INTRODUCTION

Within the DaimlerChrysler UTA (Urban Traffic Assistance) project, different vision modules for inner city traffic have been developed [1,2]. This includes fast stereo vision for Stop&Go, traffic sign and light recognition as well as pedestrian recognition and tracking. It is the goal of our current investigations to add collision avoidance capabilities to the existing system. In particular, we intend to recognize situations that implicate a high risk of accidents with children running across the road. A scooter coming from the side, a child looming between parking cars as shown in Fig. 1.1, or a ball bouncing on the road indicate such dangerous situations. A warning as well as an emergency reaction have to take place instantaneously in order to prevent accidents with serious injuries.

Relevant objects must be detected and classified in real-time from the moving car. For obstacle recognition, we generally use stereo analysis followed by a classification stage.

Stereo vision delivers three-dimensional measurements. A height threshold is applied in order to distinguish between ground and obstacle features. Points above ground are grouped to objects. Detected objects are tracked over time to estimate their motion.

Fig. 1.1 A child behind a car

Although very powerful, stereo analysis has three drawbacks with respect to the application that we have in mind. First, the grouping process tends to merge objects which are close to each other, e.g. a pedestrian in front of a vehicle or a child behind a car. Secondly, the height threshold implies the risk to miss small obstacles which are close to the ground. Thirdly, motion information included in the sequence is exploited for the detected objects only.

Motion analysis, on the other hand, allows to estimate the motion of any pixel based on the analysis over time and thus detection of any moving object.

In vehicles, a precise recovery of the ego-motion is necessary in order to distinguish between static and moving objects. Unfortunately, the ego-motion estimation is a difficult problem which requires considerable computational power and usually lacks robustness. The presence of optical flow does not automatically indicate a moving object, while zero flow does not mean a zero risk. Depending on depth, a collision could take place in either case.

A proper combination of both techniques promises the optimal exploitation of the available information in space and time. In this paper, we present an elegant method which uses the fact that stereo disparity and optical flow are connected via real-world depth. The so called "flow/depth constraint" allows to test each motion vector directly against the stereo disparity to detect moving objects. The detection works within a few image frames with very low computational cost.

In section 2 we describe the systems used for stereo and motion analysis. The fusion of stereo and motion data by means of the flow/depth constraint is presented in section 3.

A special problem is the mentioned ball since it appears very small in images at relevant distances of 20-30 meters and may still be missed by our combined detector if it moves too fast with respect to its own size. In order to reach maximum reliability, we have therefore developed an appearance based algorithm which is described in section 4.

II. Stereo and Motion

A. Stereo Vision

Our stereo analysis [3] is based on a correlation-based approach. In order to reach real-time performance on a standard PC, design decisions need to be drawn carefully.

First of all, we use the sum-of-squared (SSD) or sum-of-absolute (SAD) differences criterion instead of expensive cross correlation to find the optimal fit along the epipolar line. Wrong results due to different mean and variance of the image pairs can be avoided if gain and shutter of the cameras are appropriately controlled.

Secondly, in order to speed up the computation, we use a multi-resolution approach in combination with an interest operator. The idea is to find correspondences on a coarse level that can be recursively refined. First, a gaussian pyramid is constructed for the left and right stereo image. Areas with sufficient contrast are extracted by means of a fast vertical Prewitt edge detector.

Pixels with sufficient gradient are marked, from which a binary pyramid is constructed. A pixel (i,j) at level n is marked if one of its 4 corresponding pixels at level n-1 is set. A non-maximum suppression is applied to the gradient image in order to further speed up the processing. In this case, we find about 1100 attractive points at pyramid level zero (original image level), 700 at level one and 400 at level 2 on typical image sequences. Only those correlation windows with the central pixel marked in these interest images are considered during the disparity estimation procedure.

Depending on the application, the correlation process starts at level one or two of the pyramid. If D is the maximum searched disparity at level zero, it reduces to $(D/2)^n$ at level n. At level 2 this corresponds to a saving of computational burden of about 90% compared to a direct computation at level zero. Furthermore, smaller correlation windows can be used at higher levels which again accelerates the computation.

The result of this correlation is then transferred to the next lower level. Here, only a fine adjustment has to be

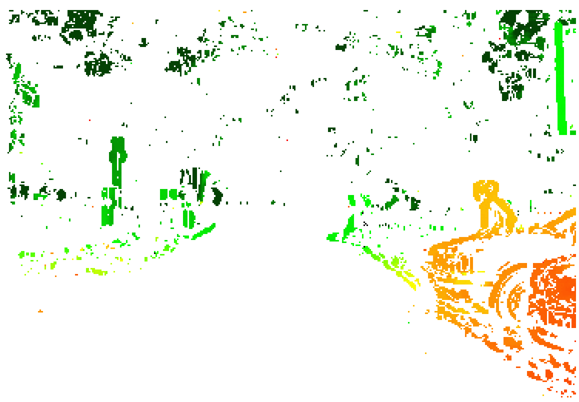

Fig. 2.1: Color encoded disparity image generated by the correlation approach. Red signals close, green means far.

performed within a small horizontal search area of +/- 1 pixel. This process is repeated until the final level is reached. At this level, subpixel accuracy is achieved by fitting a parabolic curve through the computed correlation coefficients.

The price we have to pay for this fast algorithm is that mismatches in the first computed level propagate down through the pyramid and lead to serious errors. Since the quality of a found match cannot be judged by the measured SSD or SAD, we compute the normalized cross correlation coefficient for the best matches at the highest correlation level and eliminate bad matches from further investigations. In addition, a left-right check can be applied to the disparity images on the different pyramid levels. In case of ambiguities, the best match or the match with the smaller disparity is selected. The latter strategy avoids the erroneous detection of close obstacles caused by periodic structures.

Usually, we start at level 2 (resolution 91x64 pixels) and allow a maximum disparity of 60 pixels corresponding to a minimum distance of 4 meters. In this case, the total analysis including pyramid construction runs at about 30 milliseconds on a 700 MHz Pentium III on an average. Starting at higher levels causes problems in our field of applications, since relevant structures may be lost.

Fig. 2.1 shows the disparity image that we get by this scheme for the situation of Fig. 1.1.

B. Motion Analysis

Stereo object detection usually is done by clustering disparity features to gather 3D objects. As mentioned in the introduction, this method is not sufficient if the distance between two objects is lower than a predefined threshold. Objects with a close distance will merge to a single object even if velocities vary. For a fast detection of moving objects, regardless size and distance, it is necessary to measure motion within the images directly.

Based on performance comparison of a number of optical flow techniques, emphasizing the accuracy and density of measurements on realistic image sequences [6], we are using a basic differential (gradient based) optical flow method from Lukas and Kanade [13].

The gradient based method assumes that gray values of moving objects do not change over time which is usually the case in a wide range of our environmental scenes.

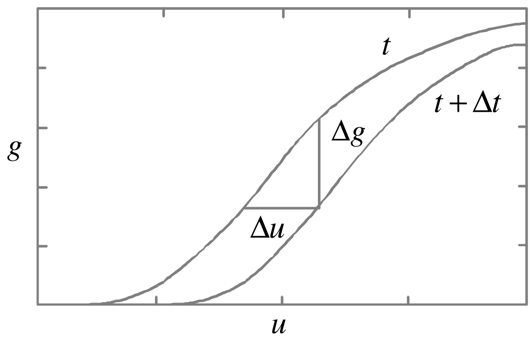

Fig. 2.2: gradient based optical flow

The computation of the optical flow is illustrated by Fig. 2.2. which is leading to the one dimensional continuity equation (2.1) where the gray value shift Δu is given as the ratio between the temporal and spatial derivatives g_t and g_u.

$$g_t + \Delta u \cdot g_u = 0 \qquad (2.1)$$

$$g_t + \Delta u \cdot g_u + \Delta v \cdot g_v = 0 \qquad (2.2)$$

Accordingly, equation (2.2) can be derived for the two dimensional case. The two dimensional optical flow (Δu , Δv) is given by the least squares solution of (2.2). And, within a small image region of N pixels, we get the following:

$$\begin{array}{ll} G_{uu} = \sum_1^N g_u^2 & G_{ut} = \sum_1^N g_u g_t \\ G_{uv} = \sum_1^N g_u g_v & G_{vt} = \sum_1^N g_v g_t \\ G_{vv} = \sum_1^N g_v^2 & G_{uu}G_{vv} - G_{uv}^2 \neq 0 \end{array} \qquad (2.3)$$

$$\begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \begin{bmatrix} \dfrac{G_{vt}G_{uv} - G_{ut}G_{vv}}{G_{uu}G_{vv} - G_{uv}^2} \\ \dfrac{G_{ut}G_{uv} - G_{vt}G_{uu}}{G_{uu}G_{vv} - G_{uv}^2} \end{bmatrix} \qquad (2.4)$$

As an example, the resulting optical flow field is shown in Fig. 2.3.

Of course, many different methods for optical flow computation like region-based matching [8], energy-based [9] and phase based [10] methods are available.

The basic gradient method can also be improved by using either second order derivatives or smoothness constraints for the flow field [11].

Fig. 2.3: computed gradient flow field

However, none of the above methods is capable of computing dense optical flow fields under real-time conditions. Usually, special hardware and parallel processing is needed in order to reach acceptable frame rates whereas the basic gradient flow can be computed in real-time on a standard PC. Furthermore, we will show that in combination with stereo, the basic method is more than sufficient for our detection problem.

III. Fusion of Stereo and Motion

Both methods, stereo and motion, have certain disadvantages for object detection. As described above, stereo extracts depth information without correlation over time. The optical flow on the other hand is able to detect even small gray value changes providing the possibility for early detection of moving objects. But with a moving camera, it lacks from suppression of background-flow without depth information.

In order to use the information of both systems in an optimal way, we suggest a sensor fusion method. We will show that with the proposed fusion of stereo and motion both methods supplement their shortcomings leading to a robust detection of arbitrary moving objects.

A. *Flow/Depth constraint*

Let us assume a purely longitudinal moving camera and a stationary environment for the moment. For the transformations between the 3D world coordinate system (x, y, z) and the corresponding 2D image coordinate system (u, v), we are using a pinhole camera model with the focal length f and s_u as the size of a sensor element of the camera chip. With the pinhole camera model and the stereo base line b, we can derive the disparity D and the optical flow $(\dot{u}, \dot{v})$ from triangulation leading to the following equations:

$$D = \frac{f \cdot b}{s_u \cdot z} \quad , \quad \frac{\dot{u}}{u} = \frac{\dot{z}}{z} \quad , \quad \frac{\dot{v}}{v} = \frac{\dot{z}}{z} \qquad (3.1)$$

Both, disparity and optical flow, depend on the real-world depth z. Therefore, the optical flow field can be computed from depth information and vice versa for stationary objects.

However, computation of the real-world depth is not necessary in our case. Switching variables for vehicle speed $\dot{z} = \Delta s$ and the horizontal and vertical components of the optical flow $\dot{u} = F_u$, $\dot{v} = F_v$, the depth factor is eliminated by building the quotient between the optical flow and the disparity. Separately applied to the horizontal and vertical components of the optical flow, this leads to the following constraints:

$$\frac{F_u}{D} = \frac{s_u \cdot \Delta s}{b \cdot f} \cdot u \qquad \frac{F_v}{D} = \frac{s_u \cdot \Delta s}{b \cdot f} \cdot v \qquad (3.2)$$

Equations (3.2) can be illustrated by inclined planes over the image region (u, v). The gradient of the planes is determined by the stereo base line b, the size of a

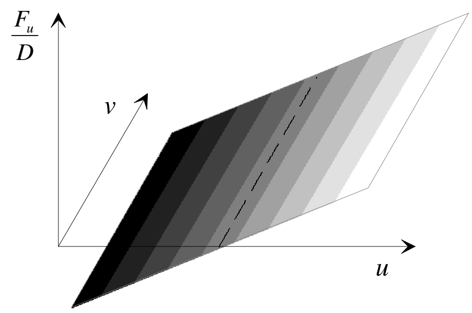

Fig. 3.1: flow/depth quotient plane

sensor element of the camera chip s_u, the focal length f and the vehicle speed Δs [m/frame]. Fig. 3.1 shows this plane for the horizontal component of equation 3.2.

Using our in-vehicle stereo camera system, the camera parameters f, s_u and b usually remain constant while only the speed varies over time. Therefore, the inclination of the plane changes as a function of the vehicle's speed only.

Fig. 3.2: deviation of flow/depth value (green) from the quotient plane (red) if a moving object is present

The quotient values for pixels belonging to stationary objects will match the plane. If the quotient does not match the value of the plane, we have to consider a moving object at this image position. Fig. 3.2 shows four consecutive images of a test sequence. All objects within the scene are stationary except one vehicle which backs into the street from the right while the camera is moving forward. The flow/depth quotient is computed for one line in the image center only. The corresponding values are displayed in green. The value of the quotient plane is displayed in red. If stationary objects are present, the quotient measurements follow the predefined value of the plane. Quotient values corresponding to the moving object vary distinctively from the plane.

B. *Quotient Noise*

As we see from Fig. 3.2, there is some measurement noise from the underlying stereo and optical flow within the flow/depth quotient which complicates segmentation of moving objects. But since the measurement noise for the disparity and optical flow preprocessing is well known, we can derive the maximum error of the quotient and use it as a threshold function for the segmentation.

From

$$Q + \Delta Q = \frac{F + \Delta F}{D + \Delta D} = \frac{F^*}{D^*} \tag{3.3}$$

we get the following function for the maximum quotient error for the horizontal and vertical flow, respectively:

$$\Delta Q = \frac{1}{D^*}[\Delta F - \Delta D Q] \tag{3.4}$$

where Q is the value of the flow/depth plane and D^* is the current measurement value for the disparity. $\Delta D, \Delta F$ are the known maximum errors for the disparity and optical flow preprocessing. Together with equation (3.2) the maximum error of the horizontal quotient value is given by:

$$\Delta Q = \frac{1}{D^*}[\Delta F - \Delta D \cdot \frac{s_u}{b \cdot f} \cdot u \cdot \Delta s] \tag{3.5}$$

Except u, equation (3.5) is the same for the vertical quotient. Fig. 3.3 illustrates the maximum allowed deviation from the plane, which basically is the absolute value of equation (3.5). We will use this as the threshold function.

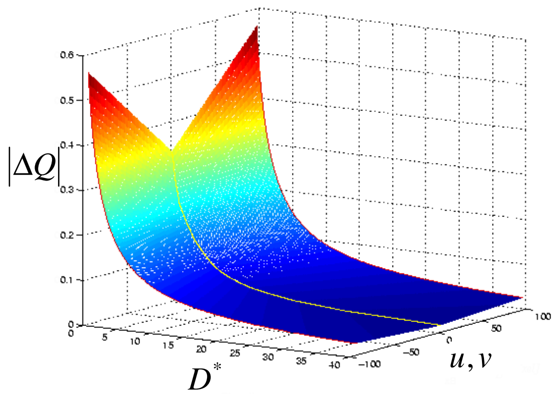

Fig. 3.3: maximum flow/depth quotient noise

Segmentation of moving objects is a three step process:

1. Compute the horizontal and vertical quotients from the optical flow and the disparity for every pixel

(u,v) for which depth and motion information is present.
2. Compare the computed quotient values with the reference values from the flow/depth plane at position (u,v) multiplied with the known vehicle speed Δs.
3. Tag image position (u,v) as "moving object" if the difference between reference value and quotient is more than $|\Delta Q|$ in at least one direction.

C. *Stabilization*

So far, pure longitudinal camera motion was assumed. We use the above method within a demonstrator vehicle where the camera is mainly moving in longitudinal direction. Additionally, there are rotational components about all three axes.

There is a distinct flow pattern corresponding to rotation and translation along every camera axes. As the camera movement is a combination of camera translation and rotation, the optical flow is a linear combination of independent components.

In order to use the flow/depth constraint as described above, we have to stabilize the image so that all rotational components are zero and only the translational flow remains.

Our stabilization is estimating self-motion using a matched filter method [12]. Each predefined filter is tuned to one flow pattern corresponding to either camera pitch, yaw or roll according rotation for the three camera axes. We assume that the flow preprocessing stage provides the optical flow as an input to the matched filters. The elimination of the rotational flow components is done in three steps:

1. Compute the filter output from the weighted sum of the scalar product between the optical flow and the matched filter pattern at each image position. This results in a single scalar which is the rotational speed for this axis.
2. An estimate for the rotational flow field is given by the product of the matched filter pattern and the rotational speed from the first step.
3. The compensated flow is given by the difference between the measured optical flow and the estimated rotational flow field from step 2.

The method is very well adapted to our stabilization task. Based on the optical flow which we take from the preprocessing stage there is only a small amount of computational power needed for the stabilization within every image cycle. The matched filter patterns for all three axes do not change over time, so they can be computed only once when the system is initialized. If we assume, that the optical flow is present for n pixels within the image, we only need $2n$ MUL, $2n-1$ SUM and 1 DIV operation to compute the rotational speed from step 1. The flow prediction from step 2 needs $2n$ MUL and the compensation from step 3 needs $2n$ SUB operations.

D. *Results*

The system has been tested on several inner city image sequences with pedestrians involved. As an example, one of these scenes is shown in Fig. 3.4.

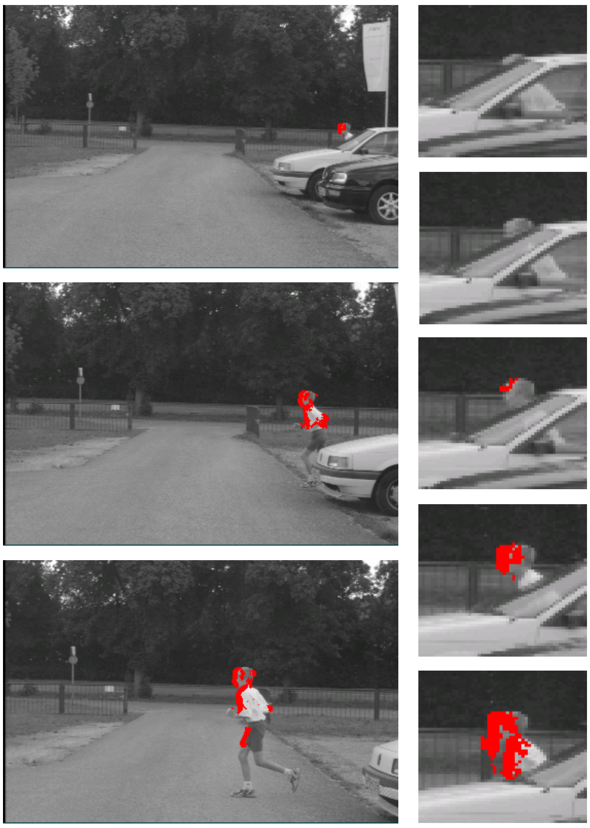

Fig. 3.4: detection of a child within an image sequence taken from a moving camera

The sequence has been taken from our in-vehicle stereo camera system. The vehicle speed is 18 km/h and the slight pitch and yaw movement of the camera has been compensated by the described matched filter method.

The result of the flow/depth constraint is overlaid onto the image. As can be seen, the algorithm is very sensitive to movements that don't match the motion of a static environment with respect to the moving camera, while background noise is very low.

The robust and fast detection can only be achieved because our fusion method is using the information from the stereo and the optical flow subsystems in an optimal way. The head of the child is detected within only three image frames after its first appearance behind the front window of the car. With single use of stereo or optical flow alone this wouldn't be possible.

The detection is independent of size or shape of the object. Since everything is done on a small pixel based neighborhood, the detection even works for non-rigid motion from pedestrians where motion varies for different parts of the body. However, in Fig. 3.4, the motion of legs and arms with respect to their size is fairly high and therefore out of the measuring range of our current motion analysis.

The flow/depth constraint also works on areas where the flow is zero. Due to the fact that our camera is moving, a flow equals zero does not automatically mean a zero risk. The subimages in Fig. 3.4 have been cropped from the original video at a fixed position. Because the camera is moving, there is obvious motion of stationary objects in the background and the cars in front. Even though the child is moving with respect to world coordinates, there is almost zero optical flow for the child's head since its position within the image stays nearly constant over time. But as one can see there is no difference in detection even under this extreme conditions.

The current system works for low vehicle speed. Due to our optical flow algorithm, the range for valid image motion is restricted to ± 2 pixel/frame. With the current camera and a video rate of 25 frames/s, this restricts the maximum vehicle speed to 25km/h. As flow range is limited, the used stabilization is optimal for small rotational velocities only.

In order to overcome this restrictions, we are working on a multi-scale approach for optical flow which will extend the current measurement range.

IV. BALL DETECTION

If a ball bounces on the road, the risk of a child following is very high. Therefore, we have to pay special attention to the recognition of this situation.

The power of human perception stems from its parallel processing capability. If a ball moves across the road, motion will attract our attention immediately. This is what we try to copy if we do motion analysis. However, if the motion is small, humans will recognize the ball based on its appearance.

There are two cases where the above described flow/depth analysis may fail to detect balls. First, relevant balls appear very small in the images and secondly, they can move too fast for the motion analysis. For these cases we have built an alternative appearance based detection scheme that consists of 4 steps [5]:

1. Potential balls are detected by means of a Hough transform for circles of different radii. In order to detect the ball early, we have to search for small circles.
2. Relevant balls have a certain size that can be estimated if the distance to the camera is known. This distance is measured utilizing the stereo vision capability of UTA. Circles that do not match the

Fig. 4.1: Situation with three balls and structured background. Potential balls found by the evaluation of the Hough accumulator are marked.

Fig. 4.2 Hough accumulator. The darkness is proportional to the probability of a detected circle.

Fig. 4.3: Remaining ball candidates after size and position check.

size constraint are rejected. In addition, balls outside a predefined driving corridor are ignored in the sequel.

3. Although most of the erroneously detected obstacles (false positives) can be rejected by the second stage, hub caps and other circular objects cause false alarms. In order to solve this problem, an artificial neural network (ANN) has been trained to distinguish between balls and other circular objects.
4. Finally, the motion parameters are determined and reactions to balls with physically impossible motion are disabled.

These four steps are described in detail below.

A. *Detection*

The Hough transform has proven to be a robust detection scheme in many applications. Edge points of lines, circles, ellipsoids etc. can be mapped to common points in the parameter space, even if the contours are noisy due to low contrast or low signal to noise ratio. Since balls are ideal circles if the pixels of the camera are quadratic, it is promising to use a version which is specialized to detect circles.

The representation of a circle by

$$(x-x_0)^2+(y-y_0)^2-{r_0}^2=0$$

requires three parameters, i.e. the two coordinates which specify the center (x_0, y_0) and the radius r. Therefore, the parameter space is three-dimensional. Since the radii of the circles vary from 3 pixels at 25m to 16 pixels at 5m (UTA is equipped with 12mm lenses and ½" imager), the size of the accumulator would be 384x256x14.

In order to achieve real-time capability, the basic algorithm needs to be improved. Above all, we have to avoid the third dimension as much as possible. We start with a simple but fast Sobel filter to determine the spatial derivatives which are then squared, added and thresholded. Then, a non-maximum suppression is carried out and the local orientation of each remaining edge point is calculated. The obtained edge-points are mapped onto 4 Hough spaces assuming radii of 4, 7, 10 and 13 pixels. A "max"-operator is subsequently applied to these accumulators pixel by pixel and one final "max"-accumulator is formed. Fig. 4.2 shows this "max"-accumulator for the image displayed in Fig. 4.1. Its darkness is proportional to the values of the accumulator.

Next, an adaptive threshold is applied to this accumulator and the local maxima of the obtained "blobs" are seeked. These points are good estimates for the center of potential circles. In a final step, the best center and radius is determined in the original image for each detected maximum. In the considered example, six peaks in the "max"-accumulator are above an adaptive threshold. They lead to the six detected circles shown in Fig. 4.1. Unfortunately, one in the tree right of the building is hard to see.

The parameters of this stage have been chosen to minimize the number of missed hits (false negative). Since the balls have often a low contrast we have to use low thresholds. The consequence is that we find circles that do not correspond to balls. Therefore, further computational steps are necessary to eliminate those hits.

B. *Check of size and position*

The size of balls we are looking for is typically in the range between 15 and 30 cm. If we could measure not only the radius but also the real diameter in centimeter, we would be able to eliminate many false alarms. It is straight forward to exploit UTA's stereo vision capability to determine distance and size of the potential balls.

We simply crop the image region containing the circle of interest and use a fast correlation to determine the disparity and distance of all circles. All candidates that do not meet the size expectation are deleted. In addition, circles that are outside a predefined area in the world are also removed.

Fig. 4.3 shows all circles that have been accepted in this stage. The two circles in the background have been rejected (compare Fig. 4.1).

C. *Classification*

Although stereo allows to remove many erroneous circles, many objects remain that do not correspond to balls but have the correct size like headlights, hub caps, people's heads and parts of traffic signs, just to mention some.

One might think of using simple heuristics like "balls are white circular areas". However, balls are three-dimensional objects that produce their own shadows: one half is bright and the opposite half is dark. In addition, they are often colored.

Our general approach for those problems is *training instead of programming*; therefore we regard the object

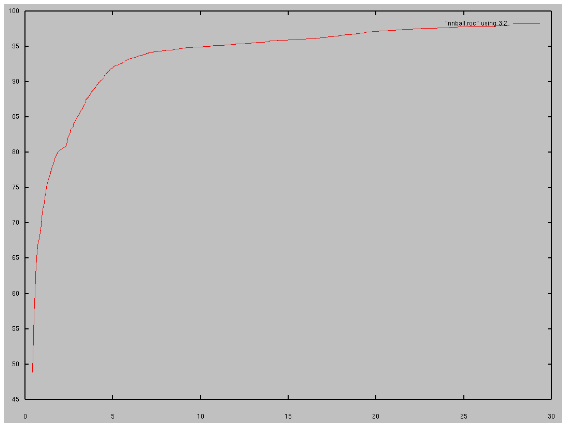

Fig. 4.5: The ROC-curve shows the percentage of correct classified balls versus the percentage of misclassification, if the final threshold is varied.

recognition problem as a classification problem to be solved by classification techniques which require a training procedure based on a large number of examples. The advantage of this approach is that no explicit models of the searched objects have to be constructed, which would often be a rather difficult, if not impossible task.

In the considered application, we use an ANN with receptive fields [4] which can be trained very efficiently and which performs very fast. The input is the cropped region of interest that has been scaled to a uniform size of 16x16 pixels. The net has two output neurons, one signaling ball and the other garbage recognition.

Robustness of the ANN classifier has been obtained by means of a bootstrapping procedures. We began with a first set of training samples, tested the resulting system in the real-world environment and retrained the recognition errors in order to generate a new version of the system that then had to undergo the same procedure, and so on.

Fig. 4.4 shows some examples of balls and garbage which have been used in the training phase. The

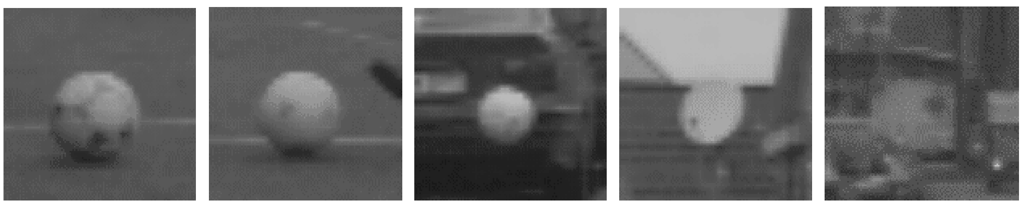

Fig. 4.4: Examples of circular objects detected by Hough transform and passed by the stereo module

Fig. 4.6: Remaining balls after classification. All garbage has been removed.

performance of the obtained classifier can be deducted from the receiver operating curve (ROC) shown in Fig. 4.5. This curve displays the percentage of correctly classified objects vs. the percentage of false positives if the decision threshold is varied. For example, if we want to classify 80% of all balls correctly, we have to accept that about 5% of all false positives are erroneously accepted.

In the considered example, the classifier rejected the circular structure on the girl of Fig. 4.3 successfully, as can be seen in Fig. 4.6.

D. *Tracking*

All remaining ball candidates are finally tracked and their motion parameters are estimated. These parameters allow to distinguish between moving balls and other circular objects that could not be rejected by the classification module. Hub caps are not moving while the headlights of oncoming vehicles, which often appear as bright circles, show a high longitudinal motion. Circles that show these behaviors are certainly not of interest for us.

Each circle that passed the stages described above is therefore tracked over time. We assume a linear motion and use a standard Kalman filter to estimate position and velocity of each object of interest. A weighted distance measure is used to set up the correspondence between the tracked circles and the detected candidates.

The tracking is illustrated by the sequence shown in Fig. 4.7. Three frames show the first detection, the last successful track and a frame in between. Additionally the track is shown.

The corresponding Fig. 4.8 shows longitudinal, lateral and vertical position of the ball in the vehicles coordinate system. Fig. 4.9 shows the estimates of the longitudinal (relative), lateral and vertical velocity, derived by the mentioned Kalman filter. The ball was first detected at 21m. UTA approached the ball with a speed of 11 m/sec. The lateral speed of the ball was about 2 m/sec, slowly decreasing. UTA came to a stop,

Fig. 4.7: Three images of the considered sequence showing the first detection and the last successfully tracked ball as well as one frame taken between. The track of the ball is shown additionally.

when the ball was about 7m apart.

E. Results

The system has been successfully tested many times under various weather conditions, including rain. Typically, the ball is detected at a distance of 20-25m ahead. If it shows a significant lateral motion and crosses UTA's driving corridor, the horn is activated immediately and a braking maneuver is initiated.

At a speed of 10m/sec and a moderate deceleration of 0.5g, the braking distance is about 10m. Although we have some delay in UTA's brake due to the interface we use in this experimental car, the vehicle comes to a secure stop in front of the ball.

If the speed of our vehicle would be significantly higher, the remaining braking distance would be not sufficient. However, we still contribute to the traffic safety: the horn will warn a child intending to run across the street. Additionally, the emergency braking will reduce the kinetic energy significantly and therefore diminish the impact of a potential collision.

The system processes 25 frames per second on a 700MHz Pentium. The detection stage requires about 20 msec. Stereo, classification and tracking need about 6 msec together. The system is able to detect and track several circular objects in parallel.

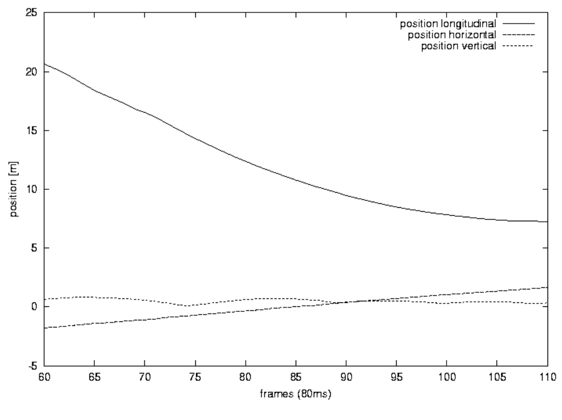

Fig 4.8: Positions of the ball obtained by tracking and Kalman filtering

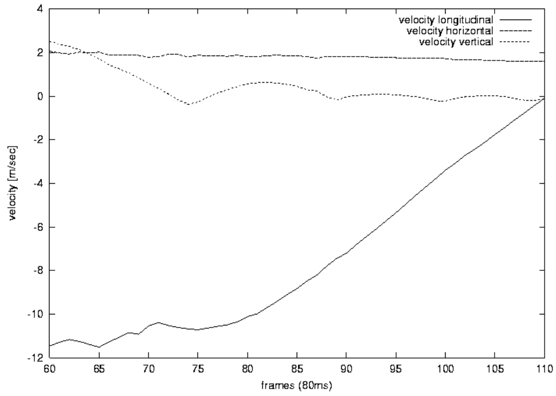

Fig 4.9: Velocities of the ball obtained by tracking and Kalman filtering (velocities relative to the braking vehicle).

V. SUMMARY

The early detection of dangerous situations in urban traffic is a serious challenge for image understanding systems. Up to now, we had stereo vision to detect obstacles in front of the car only.

The presented fusion of stereo and motion analysis is a new powerful scheme that allows early detection of moving obstacles even if they are partially occluded and non-rigid. The disparity information is already available in UTA and the simple motion analysis runs in real-time too. Since the fusion algorithm has to compare the flow/depth quotient against a threshold function at distinct points only, it is computationally highly efficient. Its current limitation to low vehicle speed due to the used optical flow measurement shall be overcome by means of a multi-scale approach.

Human drivers associate a ball bouncing on the road with a dangerous situation. The presented vision system is able to recognize those situations. Although the considered objects are quite small, the combination of Hough transform, stereo vision, classification and tracking results in a robust and fast algorithm.

VI. REFERENCES

[1] U.Franke, D.Gavrila, S.Görzig, F.Lindner, F.Paetzold, C.Wöhler: „Autonomous Driving Goes Downtown", IEEE Intelligent Systems, Vol.13, No.6, Nov./Dec.1998, pp.40-48

[2] U.Franke, D.Gavrila, A.Gern, S.Goerzig, R.Janssen, F.Paetzold and C.Wöhler: „From door to door – principles and applications of computer Vision for driver assistant systems", in

Intelligent Vehicle Technologies: Theory and Applications, Arnold, 2001
[3] U.Franke: „Real-time Stereo Vision for Urban Traffic Scene Understanding“, IEEE Conference on Intelligent Vehicles 2000, October, Detroit
[4] C. Wöhler, J. K. Anlauf. An Adaptable Time Delay Neural Network Algorithm for Image Sequence Analysis. IEEE Transactions on Neural Network, vol. 10, no. 6, pp. 1531-1536, 1999.
[5] U.Franke, A.Joos, B.Aguirre:"Early Detection of potentially harmful situations with children", Intelligent Vehicles 2001, Tokyo, Mai 2001
[6] J.L. Barron, D.J.Fleet, S.S. Beauchemein: "Performance of Optical Flow Techniques", International Journal of Computer Vision 1, 1994
[7] W.B. Thompson and Ting-Chuen Pong: "Detecting Moving Objects", Int. Journal of Comp. Vision 4, 1990
[8] P. Anandan; "A computational framework and an algorithm for the measurement of visual motion", Int. Journal of Comp. Vision 2, 1989
[9] D.J. Heeger: "Optical flow using spatiotemporal filters", Int. Journal of Comp. Vision 1, 1988
[10] D.J. Fleet and A.D. Jepson: "Computation of component image velocity from local phase information", Int. Journal of Comp. Vision 5, 1990.
[11] H.H. Nagel: "Displacement vectors derived from second-order intensity variations in image sequences", Comp. Graph. Image Processing 21, 1983
[12] M.O. Franz: "Minimalistic Visual Navigation", VDI Reihe 8 Nr. 739, 1999
[13] B. Lucas and T. Kanade: "An iterative image registration technique with an application to stereo vision", Proc. 7th Int. Conf. On Artificial Intelligence, 1981