Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Near Zone Pedestrian Detection using a Low-Resolution FIR Sensor

J.-E. Källhammer, D. Eriksson, G. Granlund, M. Felsberg, A. Moe, B. Johansson, J. Wiklund, and P.-E. Forssén

Abstract— **This paper explores the possibility to use a single low-resolution FIR camera for detection of pedestrians in the near zone in front of a vehicle. A low resolution sensor reduces the cost of the system, as well as the amount of data that needs to be processed in each frame. We present a system that makes use of hot-spots and image positions of a near constant bearing to detect potential pedestrians. These detections provide seeds for an energy minimization algorithm that fits a pedestrian model to the detection. Since false alarms are hard to tolerate, the pedestrian model is then tracked, and the distance-to-collision (DTC) is measured by integrating size change measurements at sub-pixel accuracy, and the car velocity. The system should only engage braking for detections on a collision course, with a reliably measured DTC. Preliminary experiments on a number of recorded near collision sequences indicate that our method may be useful for ranges up to about 10m using an 80×60 sensor, and somewhat more using a 160×120 sensor. We also analyze the robustness of the evaluated algorithm with respect to dead pixels, a potential problem for low-resolution sensors.**

I. INTRODUCTION

In recent years much effort has been devoted to pedestrian detection using car mounted cameras, see e.g. [1] for a review. The purpose of these systems is typically either *early warning* to alert the driver, or *injury mitigation* when a collision with the pedestrian cannot be avoided. In early warning systems the car alerts the driver, who can then decide on an appropriate evasive action. Such systems are already available on the consumer market, e.g. the Intelligent Night Vision System from Honda, but improvement of sensors and detection algorithms for early warning systems [2], [3], [4], [5] are still active research topics. Injury mitigation systems on the other hand, are still research only. They aim for pedestrian detection at a closer range, where a collision is deemed imminent, and the car should autonomously invoke countermeasures. The injury mitigation approach is motivated by studies that show that even a moderate reduction in impact speed significantly reduces the risk of fatal injury. For instance, fatal injury at a speed of 50 km/h is 5 times more likely than at 30 km/h [6].

One popular approach to pedestrian injury mitigation is the development of *time-of-flight sensors* that directly measure distance to objects in the scene. Examples of such sensors are the PMD by Audi Electronics [7], the EPT by Canesta Inc. [8], and the 3D Sensor by Siemens AG [9]. Another approach, advocated in [10] and pursued by BMW [11], is to integrate the evidence from multiple sensors such as radars, far IR, grey-scale cameras, and laser scanners.

In this paper we will investigate an alternative approach to pedestrian injury mitigation, that deploys a single low-resolution (between 80×60 and 160×120 pixel) FIR sensor. The key insight that makes this setup viable is that it allows estimation of distance-to-collision (DTC), using depth from size change and car velocity. Allowing the car to brake autonomously is a last resort, which should be invoked only when collision is inevitable. In such situations, the car will be quite close to the pedestrian (typically 5 to 15 meters away), and consequently the pedestrian will change size considerably in-between consecutive frames. The relatively large size change in-between frames is what allows us to use a low-resolution sensor. Our choice of sensor lowers the total cost of the system, and reduces the amount of data that needs to be processed in each frame.

The DTC estimation is closely related to time-to-collision (TTC) estimation [12]. Conversion of TTC to DTC requires measurements of the vehicle speed. Such measures from speedometer or ABS can be made available at low cost; in our system they are obtained through the CAN[1]-bus. The advantage with using DTC instead of TTC is that we no longer need to make the assumption that the vehicle is moving with constant speed or acceleration when integrating estimates over time, as was done in [12].

We would like to point out the differences between our system and the MobileEye one in [12]. Their scenario is to detect TTC for a car driving in front of the vehicle driving in the same direction, and suddenly braking. This problem is easier than ours (pedestrian collision-risk detection) for two reasons:

1) A car is physically larger than a pedestrian (and thus occupies a larger image area at a given distance).
2) Since the cars drive in the same direction, the actual distance for a given TTC is much shorter than for our system, and thus their car occupies an even larger sensor area than what is suggested by the TTC.

Manuscript received January 10, 2007. The authors would like to thank the Eurimus/EUREKA organization for supporting the PIMS project. We would like to thank VINNOVA (the Swedish Governmental Agency for Innovative Systems) for the support and the IVSS-program (Intelligent Vehicle Safety Systems) for the funding of the preceding Swedish PIMS project.

J.-E. Källhammer, D. Eriksson are with Autoliv Development, Wallentinsvägen 22, SE-447 83 Vårgårda, Sweden Jan-Erik.Kallhammer@autoliv.com

G. Granlund, M. Felsberg, A. Moe, B. Johansson, J. Wiklund are with the Computer Vision Laboratory, Linköping University, SE-581 83 Linköping, Sweden {gosta,mfe}@isy.liu.se

P.-E. Forssén is with the Department of Computer Science, University of British Columbia, 2366 Main Mall, Vancouver, B.C., Canada, V6T 1Z4 perfo@cs.ubc.ca

[1] Controller Area Network bus, ISO 11898.

1-4244-1068-1/07/$25.00 ©2007 IEEE.

This means that the accuracy measures given in the MobileEye paper cannot be directly compared to ours.

A. *Organization*

This paper is organized as follows: We start by giving an overview of the system components in section II. We then describe the geometry of our setup, and the principles of depth from size-change in sections III and IV. This is followed by an experimental evaluation of the accuracy of the described algorithm, and the number of dead sensor-pixels that can be tolerated, in section V. Finally, some concluding remarks and outlooks for the future are given in section VI.

II. SYSTEM OVERVIEW

In brief, our system consists of a *detection unit* that detects image regions of interest and keeps track of the horizon for image stabilization purposes, and an *estimation unit* that measures distance-to-collision (DTC) using size change and known velocity, see figure 1.

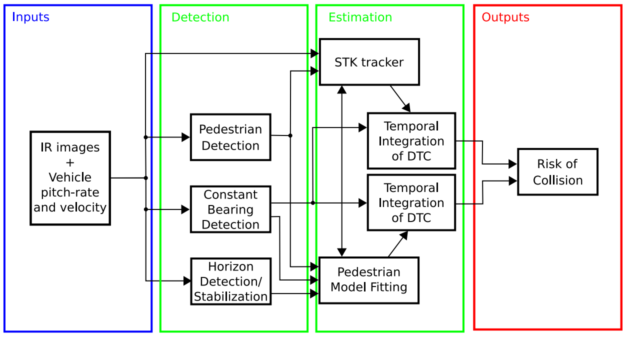
Inputs
Detection
Estimation
Outputs
IR images + Vehicle pitch-rate and velocity
Pedestrian Detection
Constant Bearing Detection
Horizon Detection/ Stabilization
STK tracker
Temporal Integration of DTC
Temporal Integration of DTC
Pedestrian Model Fitting
Risk of Collision

Fig. 1. System overview

The detection unit employs a *hot-spot detector* and a *constant bearing detector* for selecting interesting regions for further analysis. The hot-spot detector is motivated by the well known, and often used fact that pedestrians are typically warmer than their surroundings in IR images [13], [2]. The constant bearing detector makes use of the sailor principle that objects on a collision course are at a near constant bearing and decreasing range. The interesting regions found by these detectors provide seeds to the *pedestrian model fitting* box, which in the evaluated algorithm contains an energy minimization algorithm that fits a pedestrian model to the image neighborhood. The estimation unit integrates sub-pixel size-change measurements from the *STK-tracker*, a gradient based sub-pixel precision tracker, and from consecutive outputs of the energy-minimization algorithm, and combines these with car velocity readings to obtain reliable DTC estimates.

III. GEOMETRY

A. *Camera geometry*

We are currently using a camera with 55° FoV, which is well within the requirements from road geometry. We intend to base the final camera specifications on accident statistics, e.g. from the GIDAS data base [14]. This analysis will be described in a separate paper, currently under preparation.

The current camera has a 320×240 sensor, which is downsampled to give another two resolutions in our experiments: 80×60, 160×120. In our final setup we plan to reduce the vertical resolution even further.

B. *DTC estimation*

In a monocular image sequence with axial motion, one of the strongest cues to distance is size change over time. If an object increases its size in the image, its relative distance is decreasing. We can however not distinguish between size change due to car motion, and size change due to pedestrian motion, and any mix of the two. Thus, instead of estimating the current distance to the pedestrian, we choose to estimate the distance to the point of collision, i.e. the *distance-to-collision* (DTC).

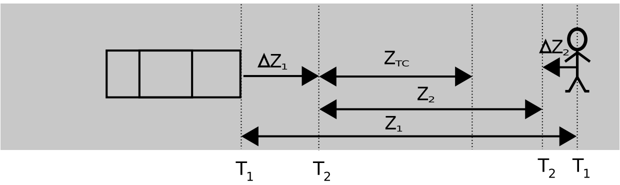

Fig. 2. Illustration of the distance-to-collision, Z_{TC}.

A 3D object which is parallel to the image plane has a length L, which is projected into the image as a length l, depending on distance Z, and focal length f:

$$l = fL/Z\,. \qquad (1)$$

At two time instances, T_1 and T_2, we have the object distances Z_1 and Z_2 (see figure 2) and get the observations:

$$l_1 = fL/Z_1 \quad \text{and} \quad l_2 = fL/Z_2. \qquad (2)$$

The image scaling $s = l_2/l_1$ becomes

$$s = l_2/l_1 = Z_1/Z_2 = (Z_2 + \Delta Z_1 + \Delta Z_2)/Z_2\,. \qquad (3)$$

We now solve for the distance Z_2, and get

$$Z_2 = (\Delta Z_1 + \Delta Z_2)/(s-1)\,. \qquad (4)$$

The time-to-collision (in number of frames) at time T_2 is given by $T_{\mathrm{TC}} = Z_2/(\Delta Z_1 + \Delta Z_2)$, and the distance-to-collision is the TTC times the car speed ΔZ_1

$$Z_{\mathrm{TC}} = T_{\mathrm{TC}} \cdot \Delta Z_1 = \frac{Z_2}{\Delta Z_1 + \Delta Z_2} \cdot \Delta Z_1\,. \qquad (5)$$

If now insert the expression for Z_2, see (4), we get:

$$\boxed{Z_{\mathrm{TC}} = \frac{\Delta Z_1}{s-1}\,,} \qquad (6)$$

see also figure 2 for an illustration.

C. Horizon detection and stabilization

An accurate estimate of the horizon position in the image serves two purposes. First of all, changes in the horizon position allows us to improve the prediction of pedestrian locations across frames, and this improves tracking performance on uneven roads. Secondly, the horizon estimate is used for verification of pedestrian hypotheses, by requiring that the horizon line must intersect each pedestrian outline.

A combination of image data and pitch rate from the CAN-bus is used to estimate the horizon. We need the image data, since the pitch rate only contains information to estimate the change in position of the horizon, not the absolute position.

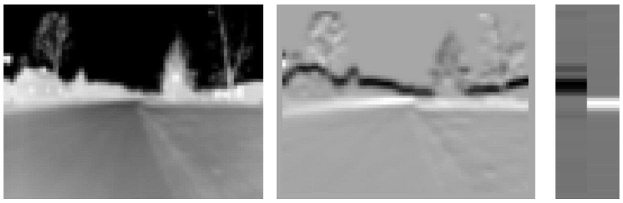

Fig. 3. Example of projected horizontal edges. Left: Original image. Middle: Derivative in the vertical direction (white indicates positive and black negative). Right: Projected negative and positive vertical derivative.

By *intrinsic horizon* we mean the position of the horizon in the image when the vehicle is moving parallel to a flat ground. We propose that the intrinsic horizon, h_0, be estimated from image data using projected horizontal edges, see rightmost image in figure 3. The projected edges are accumulated over a longer period of time, and the largest peak after smoothing is detected as the intrinsic horizon.

The change in horizon, $dh(t)$, depends on both the translation in the z-direction and the pitch rotation α. However, we will neglect translation (and acceleration) here, since the translation is small relative to the distance to the surroundings. The change in horizon will thus be approximated by a rescaling of the pitch rate.

The instantaneous horizon is the sum of the intrinsic horizon, h_0, and the change in horizon, $dh(t)$. However, the pitch rate sensor may contain a bias which when accumulated over time gives a very large error in the estimation of the horizon change. Furthermore, it is not always desirable to use the accumulated change from the pitch rate as the estimate. Rather, it is the accumulated change relative to the ground that should be used. For example, the estimated horizon should adapt to a sloping road, i.e. resume to the "resting position" h_0. Therefore, the instantaneous horizon is estimated as

$$h(t) = h_0 + dh(t, c)\,, \tag{7}$$

where

$$dh(t, c) = c \cdot \left(dh(t - T, c) + \frac{H}{FOV} \cdot \frac{d\alpha}{dt} \cdot T \right)\,, \tag{8}$$

where H is the vertical resolution, *FOV* the vertical field of view, and T the sampling rate. This relationship is basically a recursive filter, with a built-in conversion of pitch rotation measurements to image displacements. The scalar $c < 1$ controls the adaptation rate, e.g. the difference between a momentary bump and a lasting slope.

The change in horizon is used in the experiments below to improve the prediction to the tracker. A better prediction gives a better initial box position in the optimization, and the algorithm is less likely to get stuck in a local optima. The horizon position is attached to each image in the sequence, and is used as an adaptive point of origin in the tracker. This is much more computationally efficient than to resample the image.

IV. MODEL FITTING AND TRACKING

The system makes use of two units for size and size-change estimation. These are a *pedestrian model fitting* unit and a *STK-tracker* unit. Detection and tracking of pedestrians are implemented through interaction of these two units, which are described in this section.

A. Pedestrian model fitting

At the available resolutions, any approach to shape model fitting, see e.g. [15], has to use as few parameters as possible in order to obtain reliable estimates. To create such a minimal model, we start from the following assumptions:

1) The pedestrian is warmer than the environment at most parts of its contour.
2) The pedestrian is roughly shaped as in figure 4 with variable height and width; we have a rough idea about the aspect ratio though.

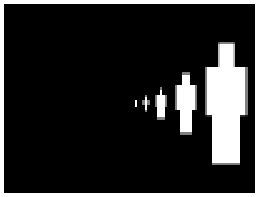

Fig. 4. Example of static shape model of the pedestrian at different sizes.

From the first assumption we conclude that the model area should have a higher average intensity than the rest of the image or at least that the model boundary is mostly brighter than the background. From the second assumption we obtain the model area $\Omega(x_0, y_0, b, h) = \Omega(\mathbf{p})$, where (x_0, y_0) indicates the position of the model and b and h its width and height, respectively. We combine the assumptions in mathematical terms by the following energy functional:

$$\begin{aligned} E(\mathbf{p}) =& f_{\mathrm{N}}|\Omega(\mathbf{p})| - \lambda_{\mathrm{DF}} \sum_{(x,y)\in\Omega(\mathbf{p})} f(x,y) + \\ &+ \lambda_{\mathrm{AR}} \left(\frac{h}{b} - r\right)\left(\frac{1}{r} - \frac{b}{h}\right) bh \end{aligned} \tag{9}$$

where $f(x, y)$ indicates a linear feature of the image (e.g. its intensity or the response of the Laplace operator), f_{N} indicates a normalization feature level, r is the model aspect ratio (here: 4), and the λs are meta-parameters that balance the three terms in the functional. (here: $\lambda_{\mathrm{DF}} = 3/2$ and $\lambda_{\mathrm{AR}} = 1/4$). The choice of an energy functional for the model fitting step was made since it has shown good performance on related problems [16].

The last term in (9) can be rewritten as

$$\frac{\lambda_{\mathrm{AR}}}{r}(h - rb)^2$$

and we know that this term is greater than or equal to 0, i.e., it penalizes deviations from our predefined aspect ratio.

We minimize (9) formally by setting the partial derivatives to zero:

$$\nabla_{\mathbf{p}} E = 0\ . \qquad (10)$$

In practice, we re-parameterize the model to use left, right, upper, and lower boundary:

$$\tilde{\mathbf{p}} = (x_0 - b/2,\, x_0 + b/2,\, y_0 - h/2,\, y_0 + h/2)^T \qquad (11)$$

and we compute discrete approximation of the derivatives w.r.t. to these bounds by finite differences. Using these derivatives, we perform a greedy-type of subspace gradient descent by moving only that boundary which has the largest effect on (9). Just like in the balloon model [15], we move the boundaries with a fixed, successively decreasing step length. This results in a very efficient implementation as we avoid interpolation until we enter the sub-pixel accuracy range. The method is iterated until the model energy cannot be further reduced with the current step-length. Finally, we recover the original parameter vector $\mathbf{p}$ such that we get the estimated center and size of the model. We can estimate the scale change from either width or height or both.

B. STK tracker

To estimate image scaling measures required by the DTC estimation described in section III-B, we employ a sub-pixel precision region tracker. The tracker is based on the assumption that the image region in one frame, $I(\mathbf{x})$, is locally identical to the one in the next frame, $J(\mathbf{x})$, after applying an unknown translation, $\mathbf{d}$, and scaling, s, i.e. $J(\mathbf{x}) = I(s\mathbf{x} + \mathbf{d})$. The method is a modification of the one described in [17]. The original paper [17] defined tracking using a full affine model, but during the experiments the authors used tracking with a translation-only model for stability reasons. The full affine model was then used for verification of the tracked region. In our system it makes sense to instead use a scale+translation model (both for tracking and verification) since we do not expect much image rotation, and we know that most things will change size in-between frames when the car is moving.

To estimate translation and scaling, we minimize the least-squares error between the image regions:

$$\epsilon = \frac{1}{N}\sum_{\mathbf{x}} \left(J(\mathbf{x}) - I(s\mathbf{x} + \mathbf{d})\right)^2\ , \qquad (12)$$

where N is the number of pixels in the local image region. The solution $(s, \mathbf{d})$ that minimizes ϵ is found iteratively by repeating three steps:

1) Replace $(s, \mathbf{d})$ by $(s + \Delta s, \mathbf{d} + \Delta\mathbf{d})$ in (12).
2) Linearize I around the point $s\mathbf{x} + \mathbf{d}$ using a Taylor expansion, and find the optimal $(\Delta s, \Delta\mathbf{d})$ by solving the equation system $\frac{\partial\epsilon}{\partial\Delta s} = 0$, $\frac{\partial\epsilon}{\partial\Delta\mathbf{d}} = 0$.
3) Update $(s, \mathbf{d}) \leftarrow (s + \Delta s, \mathbf{d} + \Delta\mathbf{d})$.

The Taylor expansion of I in step 2 becomes

$$\begin{aligned} I(s\mathbf{x} + \mathbf{d} + \Delta s\mathbf{x} + \Delta\mathbf{d}) &\approx I(s\mathbf{x} + \mathbf{d}) + \nabla I^T(\Delta s\mathbf{x} + \Delta\mathbf{d}) \\ &= I(s\mathbf{x} + \mathbf{d}) + \mathbf{M}\begin{pmatrix}\Delta s\\ \Delta\mathbf{d}\end{pmatrix}\ , \end{aligned} \qquad (13)$$

where $\mathbf{M} = \begin{pmatrix} xI_x + yI_y & I_x & I_y \end{pmatrix}$ and $\begin{pmatrix} I_x & I_y \end{pmatrix}^T = \nabla I = \nabla I(s\mathbf{x} + \mathbf{d})$. Hence, (12) is in each step approximated by

$$\epsilon \approx \frac{1}{N}\sum_{\mathbf{x}} \left(J(\mathbf{x}) - I(s\mathbf{x} + \mathbf{d}) - \mathbf{M}\begin{pmatrix}\Delta s\\ \Delta\mathbf{d}\end{pmatrix}\right)^2 . \qquad (14)$$

It is straightforward to show that the solution $(\Delta s, \Delta\mathbf{d})$ that minimizes (14) is

$$\begin{pmatrix}\Delta s\\ \Delta\mathbf{d}\end{pmatrix} = \mathbf{G}^{-1}\mathbf{e}\,, \qquad (15)$$

where

$$\mathbf{G} = \sum_{\mathbf{x}} \mathbf{M}^T\mathbf{M} \quad \text{and} \quad \mathbf{e} = \sum_{\mathbf{x}} \left(J(\mathbf{x}) - I(s\mathbf{x} + \mathbf{d})\right)\mathbf{M}^T\,. \qquad (16)$$

The transformed image $I(s\mathbf{x} + \mathbf{d})$ and its derivatives $I_x(s\mathbf{x} + \mathbf{d}), I_y(s\mathbf{x} + \mathbf{d})$ are in each iteration computed by linear interpolation. A scale pyramid is also utilized in the implementation, starting at a coarse scale and subsequently passing the optimum to finer scales.

The iterations are stopped either when the update $(\Delta s, \Delta\mathbf{d})$ is below a given threshold, or when $\mathbf{G}$ is poorly conditioned. We use the third eigenvalue of $\mathbf{G}$ as a measure of how well conditioned (15) is. This is a generalization of the check of whether the *aperture problem* is present in a plain translation based tracker, such as [17].

C. Interaction between model fitting and tracking

In each frame, the pedestrian model fitting is used to initiate the STK tracker, and the resultant size change estimate from the STK tracker is then used to update the starting hypothesis for the pedestrian model fitting. In principle, both methods are able to track pedestrians and estimate their size changes independently. However, the STK tracker uses an image region larger than the pedestrian, which makes the tracking sensitive to drift. The accuracy of the size change estimates from the STK tracker are however often higher than those from the pedestrian model fitting. Hence, we have used the model fitting to define the tracked boxes, and the STK tracker to estimate their size change.

D. Temporal Integration

There is a great deal of accuracy to be gained by temporally integrating the size change measurements. A very simple way to do this is to multiply the size changes across an interval of frames until the product reaches above a certain threshold. That is, we replace (6) with

$$Z_{\mathrm{TC}}(t) = \frac{\sum_{k=t-N+1}^{t} \Delta Z_1(k)}{\prod_{k=t-N+1}^{t} s(k) - 1} = \frac{\ldots + \Delta Z_1(t-1) + \Delta Z_1(t)}{\ldots s(t-1)s(t) - 1} \tag{17}$$

for some number of frames N. We will refer to this as *product integration* (PI). The criterion for choosing N is that it should be the smallest value that satisfies

$$\prod_{k=t-N+1}^{t} s(k) > s_{\min} \,. \tag{18}$$

V. EXPERIMENTS

Test, validation and performance measurements have been performed using three types of image sequences; synthetic, stop-motion, and from a moving car, see figure 5.

A. Sensor simulation

The actual physical sensor has a resolution of 320×240. In order to simulate sensors of lower resolution, the frames in the sequences are downsampled to either 80×60, or 160×120, before further processing.

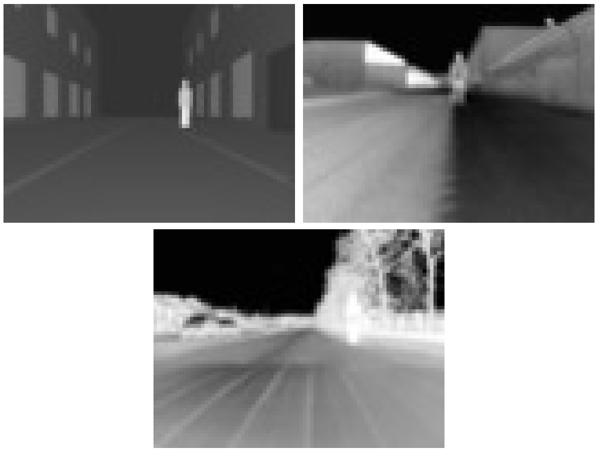

Fig. 5. Example of sequences used in the experiments. Top left (sequence 1) is a synthetically generated sequence. Top right (sequence 20) is a stop motion sequence, i.e. ground truth is known for each frame. Bottom (sequence 26) contains a pedestrian behind a speed bump. The car velocity (here around 30km/h) and pitch rate are available via the CAN bus.

Since we are using a low-resolution sensor, the effect of dead pixels is potentially more harmful than on a high resolution sensor. To demonstrate that our evaluated algorithms have a graceful degradation when the amount of dead pixels in the sensor increases, we also simulate dead pixels:

A random binary mask with a certain frequency of ones is generated. This mask is then used to set these pixels to zero, see figure 6. The mask is kept constant throughout the sequence. As a constraint, the mask is designed such that no two neighboring pixels are simultaneously dead.

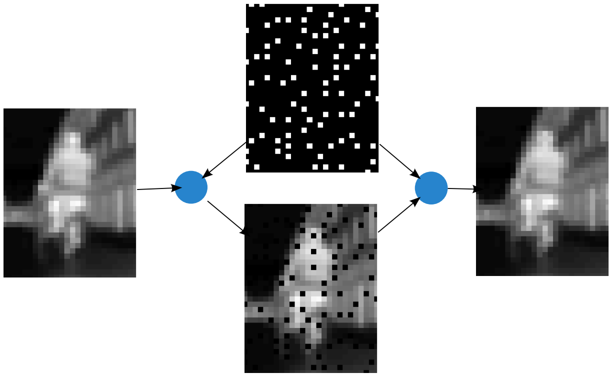

Fig. 6. Left: Original image. Center top: Pixel drop-out mask. Center bottom: Simulated raw image containing pixel drop-outs. Right: Reconstructed image. The example contains 10% drop-outs.

Dead pixels are then dealt with by interpolating them using their known locations (the mask), and their neighbors. Note that the assumption that no two neighboring pixels are simultaneously dead gives a simple reconstruction. Clusters of several neighboring dead pixels will require a more advanced reconstruction method and give larger errors.

B. Generation of ground truth

To be able to determine the DTC estimation accuracy, a reference data-set that contains the correct distance to collision in each frame, is needed. These reference data are computed by integrating car velocity data, and offsetting this by a known pedestrian distance in one of the frames. This kind of ground truth is only valid when the pedestrian stands still, or is moving in a direction perpendicular to the car motion.

Sequences generated in this way will tend to have their samples differently aligned in the DTC domain. When we later plot curves of mean DTC error and standard deviation, these measures are computed by first interpolating both ground truth and measurements to use the same DTC coordinates, and then computing the sought measures at these new points.

C. DTC accuracy and dead pixel tolerance

Several experiments with and without preprocessing (Laplace filtering) and with different parameter settings were performed to come up with a good set of parameters. During this parameter tuning, the settings were evaluated on the lowest resolution only. There may thus be a slight bias toward this resolution. The parameter tuning also made us decide on the interactive behavior of the pedestrian model fitting and tracking units described in section IV-C.

Since the results, like any set of results from less than a full scale application test, are based upon a limited set of available test data, they can only indicate trends rather than hard performance measures. Some results on the distance-to-collision accuracy are shown in figures 7, 8, 9. The individual curves are runs for 100 different simulated sensors with a 0.4% dead-pixel frequency. As can be seen, increasing the resolution increases the range for which distance estimation is stable. The effect of higher dead-pixel frequencies is demonstrated in figures 10, 11 and 12. As can be seen, the mean error stays approximately constant as the dead-pixel frequency is increased, but the variance increases.

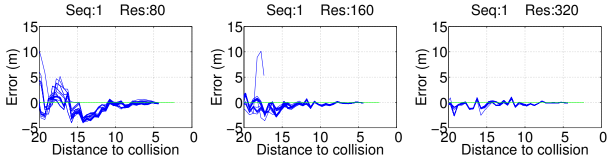

Fig. 7. The error measurements for sequence 1, with 80 × 60, 160 × 120 and 320 × 240 resolutions at a 0.4% dead pixel frequency.

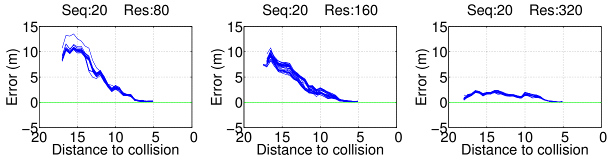

Fig. 8. The error measurements for sequence 20, with 80 × 60, 160 × 120 and 320 × 240 resolutions at a 0.4% dead pixel frequency.

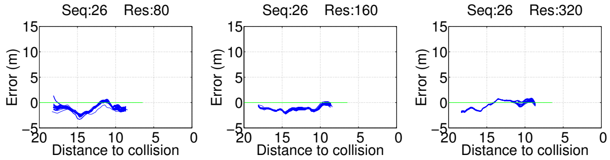

Fig. 9. The error measurements for sequence 26, with 80 × 60, 160 × 120 and 320 × 240 resolutions at a 0.4% dead pixel frequency.

The reason for the relatively poor results in sequence 20 is that the size change is consistently being underestimated, compare figures 13 and 14. There are several possible causes for this:

- Defocus may cause the image of the pedestrian to lose its perimeter, and this results in an offset in the size.
- The background is high-contrast, and the tracker may partly lock on to structures in the background. This will give a size-change in-between that of the pedestrian and the background.

Figure 15 shows the results of more quantitative experiment using simulated data. Here we have by-passed the size and size change estimations, and instead supplied the system with simulated pedestrian sizes in the image for a stationary pedestrian with height 1.8m and width 0.5m. The correct sizes have been corrupted with Gaussian noise with standard deviation $\sigma = 0.2$ pixels, which corresponds roughly to the noise levels in figures 13 and 14. The plots in figure 15 show the mean error and $\pm 3\sigma$ (i.e. 99.7% interval) curves for 100 runs. As can be seen, the error measurements from the test sequences fall well within the interval curves in figure 15.

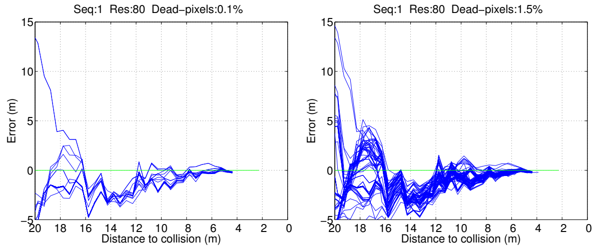

Fig. 10. Comparison of error measurements for sequence 1, having 0.1% and 1.5% dead pixels, respectively. The mean error is about the same, but the variance increases when increasing the percentage of dead pixels. See also first plot in figure 7 for result at 0.4% dead pixels.

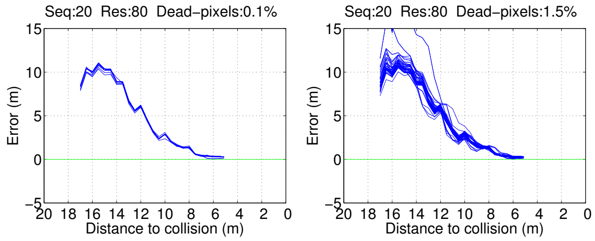

Fig. 11. Comparison of error measurements for sequence 20, also having 0.1% and 1.5% dead pixels, respectively. The main difference is that the variance increases when the percentage of dead pixels is increased. See also first plot in figure 8 for result at 0.4% dead pixels.

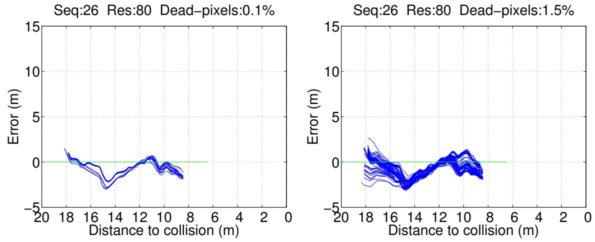

Fig. 12. The same comparison for sequence 26. Again, the mean error is about the same, but the variance increases with the frequency of pixel drop-outs. See also first plot in figure 9 for result at 0.4% dead pixels.

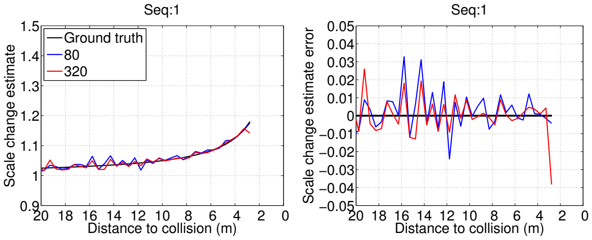

Fig. 13. Sequence 1, size-change estimate, and error.

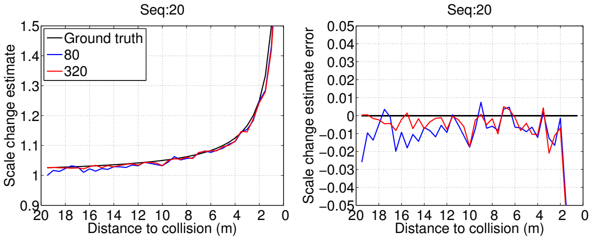

Fig. 14. Sequence 20, size-change estimate, and error.

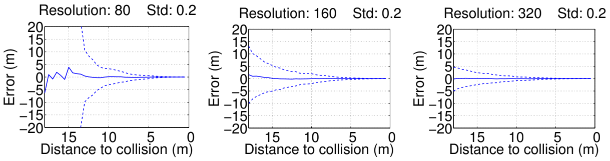

Fig. 15. Accuracy from simulation.

VI. CONCLUDING REMARKS

The reason for having a low resolution sensor is that such a system could be produced at a lower cost. This type of system is also intended to detect objects at a close range, which should permit a lower resolution.

Since the focus of this work was to determine the required sensor resolution, our experiments have focused on scenes with only one pedestrian. The detection and tracking subsystems have no problem dealing with more than one pedestrian as long as they do not occlude each other. If they do occlude each other, the system risks confusing them with one large pedestrian with variable size. This would result in an unstable size-change estimation. To avoid this potential problem, more elaborate procedures will be needed.

We have tested our method for estimation of the distance-to-collision (DTC) on a number of sequences, some results have been shown here. The preliminary tests indicate that our method may be useful for ranges up to about 10m using an 80×60 sensor, and somewhat more using a 160×120 sensor. This should be compared with the 15m detection distance reported for the PMD sensor [7], and the (at least) 16m distance reported for the 3D sensor when both car and pedestrian were stationary [9]. Both of these sensors are developed specifically for the task of pedestrian detection. The amount of test data presented in this study is still too limited to give a reliable conclusion, and further evaluation, using a laser range sensor mounted in the front of the car is under way.

References

[1] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, May 2006.

[2] F. Xu, X. Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 63–71, March 2005.

[3] I. Kallenbach, R. Schweiger, G. Palm, and O. Löhlein, "Multi-class object detection in vision systems using a hierarchy of cascaded classifiers," in *Intelligent Vehicles Symposium*. Tokyo, Japan: IEEE, June 2006.

[4] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *Intelligent Vehicles Symposium*. Tokyo, Japan: IEEE, June 2006, pp. 206–212.

[5] J.-E. Källhammer, "The road ahead for car night-vision," *Nature Photonics*, pp. 12–13, September 2006.

[6] G. J. L. Lawrence, B. J. Hardy, J. A. Carroll, W. M. S. Donaldson, C. Visvikis, and D. A. Peel, "A study on the feasibility of measures relating to the protection of pedestrians and other vulnerable road users - final report," European Commision, Tech. Rep. FIF.20030937, June 2004.

[7] B. Fardi, J. Dousa, G. Wanielik, B. Elias, and A. Barke, "Obstacle detection and pedestrian recognition using a 3d pmd camera," in *Intelligent Vehicles Symposium*. Tokyo, Japan: IEEE, June 2006, pp. 225–230.

[8] S. Hsu, S. Acharya, A. Rafii, and R. New, "Performance of a time-of-flight range camera for intelligent vehicle safety applications," in *10th International Forum on Advanced Microsystems for Automotive Applications*, Berlin, April 2006, pp. 205–219.

[9] P. Mengel, "Three-dimensional cmos image sensor for pedestrian protection and collision mitigation," in *10th International Forum on Advanced Microsystems for Automotive Applications*, Berlin, April 2006, pp. 23–39.

[10] N. Checka, "Fast pedestrian detection from a moving vehicle," in *Proceeings of the 2004 Student Oxygen Workshop*, V. Sinha, J. Eisenstein, and T. M. Sezgin, Eds., September 2004. [Online]. Available: http://sow.csail.mit.edu/2004/proceedings/Checka.pdf

[11] L. Walchshaeusl, R. Lindl, K. Vogel, and T. Tatschke, "Detection of road users in fused sensor data treams for collision mitigation," in *Proceedings of the 10th International Forum on Advanced Microsystems for Automotive Applications (AMAA'06)*. Berlin: VDI/VDE/IT, April 2006.

[12] E. Dagan, O. Mano, G. P. Stein, and A. Shashua, "Forward collision warning with a single camera," in *IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004, pp. 37–42.

[13] M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, and M.-M. Meinecke, "Pedestrian detection for driver assistance using multiresolution infrared vision," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1666–1678, November 2004.

[14] "GIDAS web site," `http://www.gidas.org/`.

[15] L. D. Cohen, "On active contour models and balloons," *Computer Vision, graphics, and Image Processing. Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.

[16] D. Cremers, S. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk." in *DAGM-Symposium*, 2004, pp. 36–44.

[17] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR'94*, Seattle, June 1994.