GENERIC MOTION BASED OBJECT SEGMENTATION FOR ASSISTED NAVIGATION

Sion Hannuna
Department of Computer Science, University of Bristol, Bristol, UK
hannuna@cs.bris.ac.uk

Xianghua Xie
Department of Computer Science, University of Wales Swansea, Swansea, UK
x.xie@swansea.ac.uk

Majid Mirmehdi and Neill Campbell
Department of Computer Science, University of Bristol, Bristol, UK
{*majid,campbell*}*@cs.bris.ac.uk*

Keywords: uncategorised object detection, stereo depth, assisted blind navigation, sparse optical flow.

Abstract: We propose a robust approach to annotating independently moving objects captured by head mounted stereo cameras that are worn by an ambulatory (and visually impaired) user. Initially, sparse optical flow is extracted from a single image stream, in tandem with dense depth maps. Then, using the assumption that apparent movement generated by camera egomotion is dominant, flow corresponding to independently moving objects (IMOs) is robustly segmented using MLESAC. Next, the mode depth of the feature points defining this flow (the foreground) are obtained by aligning them with the depth maps. Finally, a bounding box is scaled proportionally to this mode depth and robustly fit to the foreground points such that the number of inliers is maximised.

1 INTRODUCTION

This paper describe a method for isolating and annotating moving objects using head mounted stereo cameras worn by an ambulatory or stationary person. The work is carried out in the context of a large, multifaceted and EU-funded project (CASBliP[1]) which aims to develop a multi-sensor system capable of interpreting basic characteristics of some primary elements of interest in outdoor scenes (i.e. in city streets) and transforming them into a sound map for blind users as a perception and navigation aid.

One of the key tasks is to effectively detect moving objects, which may pose a danger to the user, and estimate their distance and relative motion. Unlike, for example, in autonomous vehicle navigation (de Souza and Kak, 2002; Leonard, 2007), where the camera egomotion can be estimated based on auxiliary measurements, e.g. using speedometers, our cameras can undergo arbitrary motion to six degree of freedom (dof). Such egomotion introduces significant relative motions for all objects in the scene, which makes it difficult to detect independent moving objects and even more difficult in order to do so in close to real time while simultaneously estimating the depth. Fortunately, since the speed of the user is much slower than that of the objects of interest, such as a car, the translational component of the egomotion can be neglected. Moreover, the user will be aware of their own cadence and trajectory, thus accurate estimates of true camera egomotion are unnecessary. However, it is still challenging to efficiently differentiate apparent motions induced in the scene by camera movement from that originating from real movement in the environment.

Pauwels and Hulle (Pauwels and Hulle, 2004) propose a M-estimator based robust approach to extracting egomotion from noisy optical flow. Tracked points whose trajectory do not originate in the focus of expansion (FOE) are deemed to belong to independently moving objects. This assumption fails for objects traveling in front of the camera towards the FOE. In (Badino, 2004), a method for deducing egomotion in a moving vehicle using a mobile stereo platform is described. They utilise a combination of 3D point correspondences, and a smoothness of motion constraint to deduce vehicle motion. In (Ess et al., 2007), the ground plane and pedestrians are simultaneously extracted using stereo cameras mounted on a trolley. Although they show impressive results, appearance

[1] www.casblip.upv.es

based detection is not sufficient enough for the purposes of our application. Other work that deal with the problem of segmenting independent motion with camera egomotion include (Rabe et al., 2007), (Yuan et al., 2007) and (Yu et al., 2005).

Depth estimation or disparity computation is often carried out based on the assumption that depth discontinuity boundaries collocate with intensity or colour discontinuity boundaries. The search for this collocation is based on intensity similarity matching from one image to the other, which includes stages such as matching cost computation and aggregation, disparity computation and refinement. Optimisation plays an important role in disparity estimation (Gong and Yang, 2007). Recent comparative studies, such as (Scharstein and Szeliski, 2002), have shown that graph cut (Veksler, 2003) and belief propagation (Felzenszwalb and Huttenlocher, 2006) are two powerful techniques to achieve accurate disparity maps. However, both are computationally expensive and require hardware solutions to achieve near real time performance, e.g. (Yang et al., 2006).

In section 2, we first provide an overview of the proposed method, and then elaborate each of its stages in some subsections. Then, experimental results are reported in section 3, followed with conclusion in section 5.

2 PROPOSED APPROACH

The primary aim here with respect to generic object detection is to identify objects moving independently in the scene. We are not concerned with the specific class of objects, but just to extract sufficient information for a later cognitive module to interpret if the motion of the object can pose a danger to the visually-impaired user. This is achieved by tracking a sparse set of feature points, which implicitly label moving objects, and segmenting features which exhibit motion that is not consistent with that generated by the movement of the stereo cameras. Sparse point tracking has previously been applied successfully to motion based segmentation in (Hannuna, 2007). Dense depth maps are simultaneously extracted, yielding locations and 3D trajectories for each feature point. These depth maps are also required for input into a later stage of the CASBliP project (the sonification process to generate a sound map), so they do not incur extra computational burden compared to using sparse depth maps.

The Kanade-Lucas-Tomasi (KLT) (Shi and Tomasi, 1994) tracker is used to generate this sparse set of points in tandem with the depth estimation process. This tracker preferentially annotates high entropy regions. As well as facilitating tracking, this also ensures that values taken from the depth maps in the vicinity of the KLT points are likely to be relatively reliable as it is probable that good correspondences have been achieved for these regions.

Points corresponding to independently moving objects are segmented using MLESAC (Torr and Zisserman, 2000) (Maximum Likelihood Sample Consensus), based on the assumption that apparent movement generated by camera egomotion is dominant. Outliers to this dominant motion then generally correspond to independently moving objects. Bounding boxes are fitted iteratively to the segmented points under the assumption that independently moving objects are of fixed size and at different depths in the scene.

Segmented points are aligned with depth maps to ascertain depths for moving object annotation. The mode depth of these points is used to scale a bounding box which is robustly fit to the segmented points, such that the number of inliers are maximised. To segment more than one object, the bounding box algorithm is reapplied to the 'bounding box outliers' produced in the previous iteration. This needs to be done judiciously, as these outliers may be misclassified background points that are distributed disparately in the image. However, if a bimodal (or indeed multimodal) distribution of foreground depths is present, objects will be segmented in order of how numerously they are annotated at a consistent depth. In the current real-time implementation of the navigation system, bounding boxes are only fitted to the object which is most numerously annotated at a consistent depth. This is, almost without exception, the nearest object. Only processing the nearest object simplifies the sound map provided to user, thus providing only the most relevant information and making the audio feed easier to interpret. For example, in a scene where there are two objects present at different depths, the one with the greater number of tracking points will be robustly identified first, assuming both sets of tracking points demonstrate similar variance in their depth values.

2.1 Depth estimation

In order to estimate the distance of objects from the user in an efficient manner, a stereo grid with two cameras is used since it is generally faster than single camera temporal depth estimation. The intrinsic and extrinsic parameters of the two cameras are pre-computed using a classic chart-based calibration technique (Zhang, 1998). Sparse depth estimation, e.g. correlation based patch correspondence search and

reconstruction, is usually computationally efficient. However, it is not desirable in our application since it often results in isolated regions even though they may belong to a single object which makes it difficult to sonify. In recent years, there has been considerable interest in dense depth estimation, e.g. (Scharstein and Szeliski, 2002). We have experimented with several methods, including belief propagation (Felzenszwalb and Huttenlocher, 2006), dynamic programming (Birchfield and Tomasi, 1999), sum of absolute difference with winner-take-all optimisation (Kanade, 1994), and sum of squared differences with iterative aggregation (Zitnick and Kanade, 2000).

Although recent comparative studies, such as (Scharstein and Szeliski, 2002), suggest that scan line based dynamic programming does not perform as well as more global optimisation approaches, on our outdoor dataset it appears to be a reasonable trade-off between computational efficiency and quality (see subjective comparison in Fig. 2 in the Results section). Note that our outdoor images are considerably different from those benchmarks widely used in comparative studies. It is very common for our data to contain disparities of up to 60 pixels out of 320 pixels, which is significantly larger than most standard ones. Additionally, the variation of disparities is large, i.e. for most of the frames the disparity covers most levels from 0 to 60. A typical dynamic programming approach is the scanline-based 1D optimisation process. We follow (Birchfield and Tomasi, 1999) to define the cost function to minimise while matching two scanlines as:

$$\gamma = N_o\kappa_o - N_m\kappa_r + \sum_i \bar{d}(x_i, y_i), \qquad (1)$$

where N_o and N_m are the number of occlusions and matches respectively, κ_o and κ_r are weightings for occlusion penalty and match reward respectively, and function $d(.)$ measures the dissimilarity between two pixels x_i and y_i. For this dissimilarity measure, one that is insensitive to image sampling is used:

$$d(x_i, y_i) = \min\left\{\bar{d}(x_i, y_i, I_L, I_R), \bar{d}(y_i, x_i, I_R, I_L)\right\}, \quad (2)$$

where $\bar{d}$is defined as:

$$\bar{d}(x_i, y_i, I_L, I_R) = \min_{(y_i - \frac{1}{2}) \le y \le (y_i - \frac{1}{2})} |I_L(x_i) - \hat{I}_R(y)|, \quad (3)$$

where $I_L(x_i)$ and $I_R(y_i)$ are intensity values for x_i in the left scanline and y_i in the right scanline respectively, and $\hat{I}_R$ is the linearly interpolated function between the sample points of the right scanline. The matching is also subject to a set of constraints, such as unique and ordering constraint which simplifies dynamic programming, and 'sided' occlusion constraint to handle untextured areas. For further details, the reader is refered to (Birchfield and Tomasi, 1999).

Figure 1: Fusion of depth map with image segmentation. 1st row: the original left image and graph cut based segmentation using colour and raw depth information; 2nd row: original depth estimation and result after anisotropic smoothing based on segmentation.

However, due to the nature of the 1D optimisation, streaking artifacts inevitably result, as well as temporal inconsistency, known as flicking effects. Since the stereo cameras are constantly moving and the scene often contains moving and deforming objects, enforcing temporal consistency does not necessarily improve results. A median filter across the scan lines is used to reduce the spatial inconsistency. More advanced approaches, such as (Bobick and Intille, 1999), can be used, however, at a computational cost we can not afford. Others, such as (Gong and Yang, 2007), require dedicated hardware solutions.

We also adopt a fusion approach using image segmentation based on the assumption that depth discontinuity often collocates with discontinuity in regional statistics. Similar ideas have been recently explored, e.g. (Zitnick and Kang, 2007). However, we use a post-fusion approach instead of depth estimation from over-segmentation[2]. Smoothing is performed within each region to avoid smudging across the region boundaries. An example result is shown in Fig. 1. This segmentation is based on graph cuts (Felzenszwalb and Huttenlocher, 2004), which offers the potential of multimodal fusion of depth, colour components and sparse optical flow to obtain more coherent segmentation. We are currently also investigating Mean Shift segmentation (Comaniciu and Meer, 2002) to determine if a faster throughput can be

[2]Image segmentation is also required as part of a subsystem in our CASBliP project, not discussed in this paper, for assistance to partially sighted users. Hence, the computational overhead of fusing depth information and image segmentation is affordable.

achieved without compromising accuracy.

2.2 DEDUCING BACKGROUND MOTION MODEL

To determine the motion of an object of interest, we use a robust approach to first determine which KLT points correspond to the background region. Specifically, homography is repeatedly used to parameterise a provisional model based on the trajectory of a subset of randomly selected points over a sliding temporal window and the most likely model retained. Outliers to the most likely model correspond to independently moving object annotation.

The robust technique used here is MLESAC (Torr and Zisserman, 2000) and is outlined in Algorithm 1. This method attempts to determine the parameters of the background's motion model, using the smallest possible subset of that data. Samples are drawn randomly and used to generate a provisional model. The most likely parameter vector, assuming the outliers are randomly distributed, is retained. For each provisional model, it is necessary to iteratively determine the mixing parameter, γ, (expected proportion of inliers), which yields the highest likelihood. Our use of MLESAC as opposed to RANSAC (random sample consensus) (Fischler and Bolles, 1981) is more expensive, but avoids the need for a predefined inlier threshold when determining a consensus.

Algorithm 1 Robustly identifying dominant model

X_i represents the KLT points for the current frame
WinDiam represens the radius of the sliding temporal window
X_i^t represents the subset of X_i, tracked successfully for current temporal window
M_{prov} represents the current model
M_{best} represents the best model
N is the number of samples required for dominant model to be selected with 0.95 probability

Identify X_i^t
for $j \leftarrow 1..N$ **do**
  Randomly select s samples, X_i^s from X_i^t
  Calculate M_{prov}, using $X_{i-WinDiam}^s$ and X_i^s
  Evaluate the likelihood, L_{prov}, for M_{prov}
  if $L_{prov} > L_{best}$ **then**
    $L_{best} \leftarrow L_{prov}$
    $M_{best} \leftarrow M_{prov}$
    Record *outliers* associated with M_{best}
  end if
end for
Record M_{best} and its' associated *outliers*

With regard to model parametrization, the direct linear transformation (DLT) may be used to calculate a matrix, $\mathbf{H}$, which transforms a set of points $\mathbf{x_i}$ from one image to a set of corresponding points $\mathbf{x'_i}$ in another image. In order to fully constrain $\mathbf{H}$, four point correspondences are required (Hartley and Zisserman, 2001) (which are not collinear). If the four points selected are background points, then the transformation matrix deduced will describe the motion of the background. The transformation

$$\mathbf{x'}_i = \mathbf{H}\mathbf{x}_i, \tag{4}$$

can also be expressed in the form:

$$\mathbf{x'}_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}, \tag{5}$$

where $\times$ is the vector cross product. If the j-th row of the matrix $\mathbf{H}$ is represented by $\mathbf{h}^{j\mathrm{T}}$ and $\mathbf{x'}_i = (x'_i, y'_i, w'_i)^{\mathrm{T}}$, this may be simplified to:

$$\underbrace{\begin{pmatrix} \mathbf{0}^{\mathrm{T}} & -w'_i\mathbf{x}_i^{\mathrm{T}} & y'_i\mathbf{x}_i^{\mathrm{T}} \\ w'_i\mathbf{x}_i^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & -x'_i\mathbf{x}_i^{\mathrm{T}} \end{pmatrix}}_{\mathbf{A}_i} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{6}$$

Decomposing $\mathbf{A}$, with SVD, produces the following factorization:

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}. \tag{7}$$

The columns of $\mathbf{V}$, whose corresponding elements in $\mathbf{D}$ are zero, form an orthonormal basis for the nullspace of $\mathbf{H}$ (Press et al., 1992). In other words, they provide a value for $\mathbf{h}$, which satisfies $\mathbf{A}\mathbf{h} = \mathbf{0}$.

2.3 FITTING A BOUNDING BOX

We assume that the outliers to the background model primarily represent the foreground object. The dimensions of the bounding box are determined by multiplying the foreground points' mode depth by a scaling factor which has been empirically determined. We utilise the mode, as opposed to the mean, as it is more robust to outliers and allows the possibility of isolating more than one object if the objects' depths correspond to a multimodal distribution.

The box is fitted to the foreground points using RANSAC as summarised in Algorithm 2. For every frame, the object's centroid is estimated using three randomly selected foreground points. The bounding box is then fitted to the foreground points, based on this estimation and the number of points lying within the bounding box. It is appropriate to use RANSAC in this case as the threshold is determined by the bounding box size.

It would be possible to simply use a single foreground point to estimate the object's centroid, but using a greater number allows for more variety in the

Figure 2: 1st row: stereo images and depth estimation based on dynamic programming using 1D optimisation; 2nd row: results obtained from belief propagation, sum of absolute differences with winner-take-all optimisation, and sum of squared differences with iterative aggregation.

Algorithm 2 Robustly bounding fitting

X_{out} represents the KLT points which are outliers to the dominant model, M_{best}
CoG_{prov} represents the current centroid for the bounding box
CoG_{best} represents the centroid yielding the greatest number of inliers
N is the number of samples required for dominant model to be selected with 0.95 probability

Calculate mode depth of X_{out}
Scale bounding box size proportionally to mode depth and weighted average of centroid estimations for previous frames.

for $j \leftarrow 1..N$ **do**
  Select s random samples, X^s_{out} from X_{out}
  Calculate CoG_{prov}: the centroid of X^s_{out}
  Count the number of inliers ρ_{prov}, for CoG_{prov}
  if $\rho_{prov} > \rho_{best}$ **then**
    $\rho_{best} \leftarrow \rho_{prov}$
    $CoG_{best} \leftarrow \rho_{prov}$
  end if
end for
Use weighted average of CoG_{best} and previous estimations and record associated inliers

centroid's position. However, using more points requires more iterations in the RANSAC algorithm, so 3 are selected as a compromise. The bounding box's size and location tends to jitter due to the random sampling and inconsistency with regard to features annotated by the KLT tracker. Hence both quantities are smoothed by utilising a weighted average of their current and previous values.

In addition to fitting a bounding box to the annotation, aligning KLT points with depth maps can generate relative velocity estimates as motion in the image plane may be scaled according to the moving objects distance to that plane. Furthermore, it could potentially allow the system to determine if objects are approaching the user. Alignment is relatively simple since as the cameras are calibrated, the original images and the depth maps can be rectified and aligned based on calibration parameters.

3 RESULTS

In Fig. 2 we illustrate a comparison of several depth estimation techniques, e.g. belief propagation, sum of absolute differences with winner-take-all optimisation, and sum of squared differences with iterative aggregation. We selected the method presented in (Birchfield and Tomasi, 1999) for reasonable accuracy as a trade-off for computational efficiency for reasons as described in section 2.1.

Fig. 3 illustrates the annotation and bounding box fitting process. From left to right the images show: the KLT points, the points segmented into dominant motion (red) and outliers (blue), the depth map aligned with these segmented points, with bounding box fitted and the right shows the final annotation. Note that the depth map is smaller as it only includes portions of the scene captured by both cameras. This process is also illustrated in Fig. 4. Note that depth maps are produced concurrently with the sparse point segmentation.

Figures 5, 6 and 7 are examples of the algorithm's output. In each figure every fourth frame is shown starting top left and finishing bottom right. Pairs of

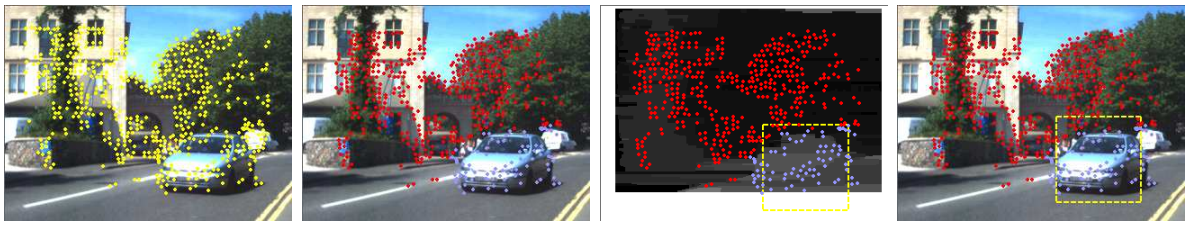

Figure 3: Left to right: sparse point tracking, foreground background segmentation and alignment with depth map, and final annotation.

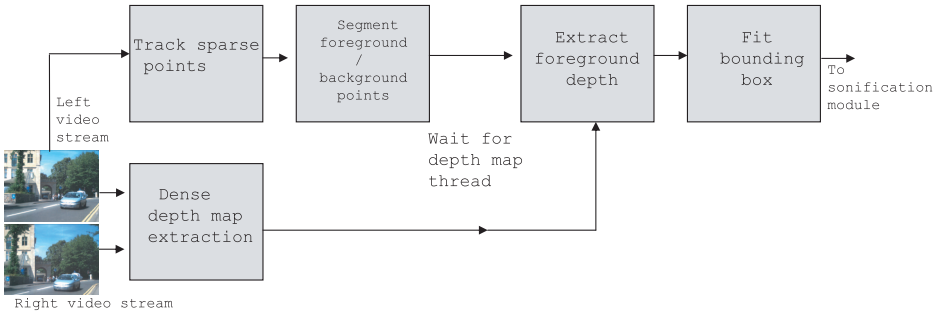

Figure 4: Flow chart illustrating bounding box algorithm and relationship with depth map extraction

images represent each frame of the sequence with the upper image illustrating the segmentation and depth maps and the lower the final annotation.

4 DISCUSSION

The system currently runs at a rate of around 8fps on a 2.39GHz laptop with 3GB RAM working on 320×240 images, when dealing with a single dominant object only; hence further refinement is necessary to be able to handle multiple objects in real-time.

We have come across no previous works with similar setups to ours, which deal with hazard detection in the outdoors using wearable cameras for blind users, to compare our results with. There are other works with a similar application in mind, e.g.: (Andersen and Seibel, 2001) in which a system is very briefly described for users with low vision, rather than no vision at all, thus relying on some user ability to see and interpret the scene; and (Wilson et al., 2007) in which a tactile-input wearable audio system is described. Also, comparisons are not easy to establish as other systems have proprietary software and hardware or require significant software redevelopment. There are also advanced works on pedestrian detection or vehicle detection, but these are mainly highly fine-tuned towards those specific classes of objects, whereas our work is simply looking for any unspecified moving objects in the scene to report audio warnings to blind users. Furthermore, our system is designed to be liberal in determining hazardous objects as it is important to report a safe object as unsafe, than vice versa. Hence, it is deemed reasonable for our bounding box to be larger and a less accurate tight fit.

5 CONCLUSIONS

A robust approach to annotating independently moving objects using head mounted stereo cameras has been proposed. This is intended for use as part of an audio-feedback system for reporting potentially hazardous objects to a blind user navigating in outdoor cityscape settings. Generic object detection is performed, without any specific object classification, to ensure fast enough performance to allow practicality of use. The system offers near real-time performance and is robust enough to tolerate the unpredictable motions associated with head mounted stereo cameras. Future work will focus on improving performance using accelerated feature tracking on the GPU.

Acknowledgements

This work was funded by EU-FP6 Project CASBliP no. 027083 FP6-2004-IST-4.

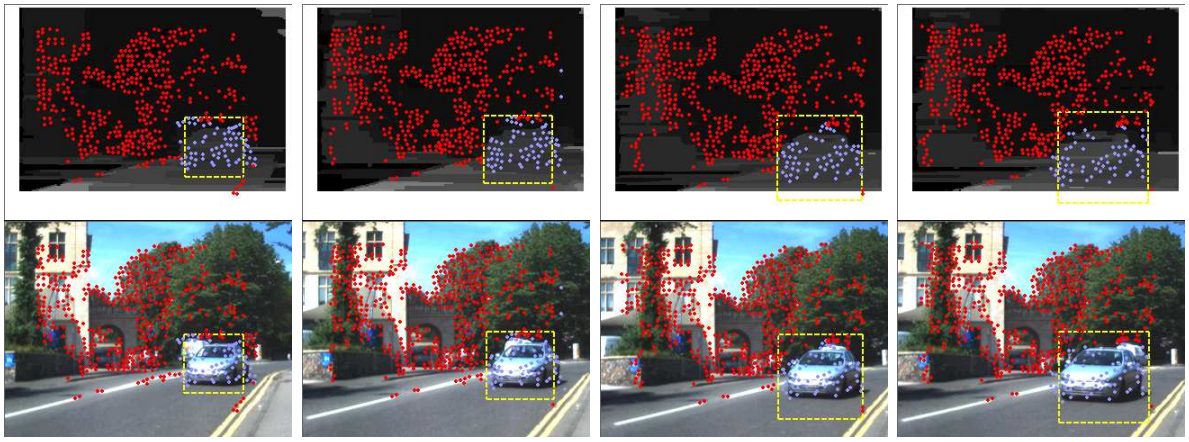

Figure 5: Every fourth frame of car approaching the cameras (depth maps and original frames are paired vertically).

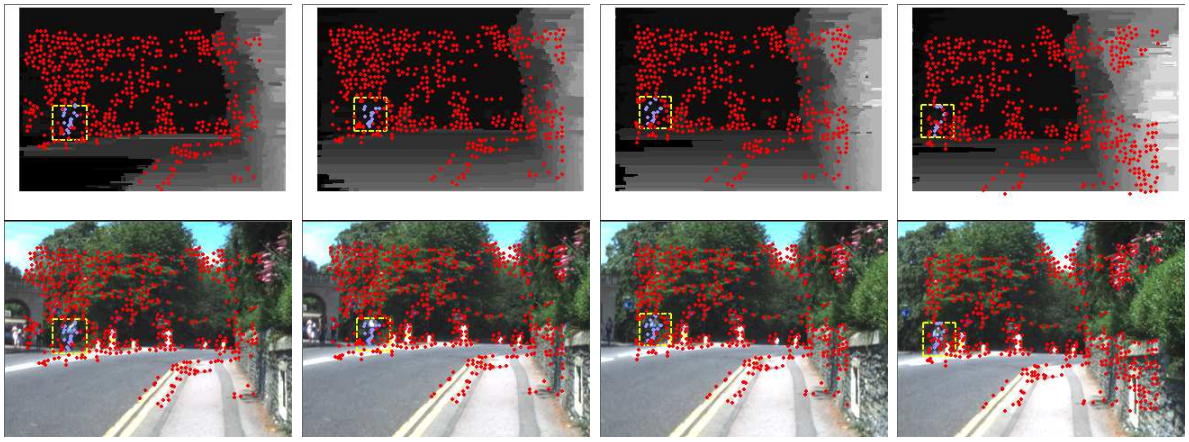

Figure 6: Every fourth frame of cyclist moving away from the cameras (depth maps and original frames are paired vertically).

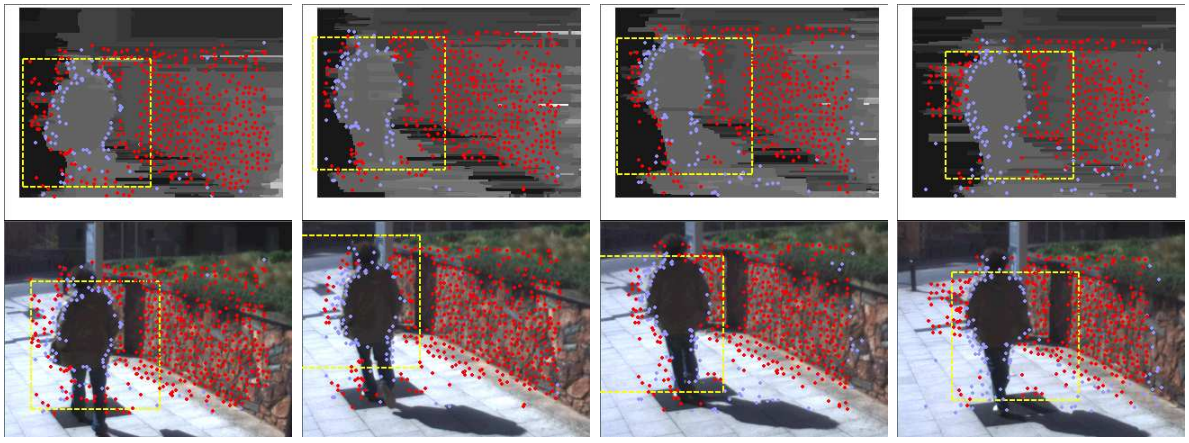

Figure 7: Every fourth frame of a person walking in the same direction as camera egomotion (depth maps and original frames are paired vertically).

REFERENCES

Andersen, J. and Seibel, E. (2001). Real-time hazard detection via machine vision for wearable low vision aids. In *ISWC '01: Proceedings of the 5th IEEE International Symposium on Wearable Computers*, page 182. IEEE Computer Society.

Badino, H. (2004). A robust approach for ego-motion estimation using a mobile stereo platform. In *1st Internation Workshop on Complex Motion (IWCM04)*, volume 3417, pages 198–208.

Birchfield, S. and Tomasi, C. (1999). Depth discontinuities by pixel-to-pixel stereo. *Int. J. Comput. Vision*, 35(3):269–293.

Bobick, A. and Intille, S. (1999). Large occlusion stereo. *Int. J. Comput. Vision*, 33(3):181–200.

Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5):603–619.

de Souza, G. and Kak, A. (2002). Vision for mobile robot navigation: A survey. 24(2):237–267.

Ess, A., Leibe, B., and van Gool, L. (2007). Depth and appearance for mobile scene analysis. In *ICCV07*, pages 1–8.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. *Int. J. Comput. Vision*, 70(1):41–54.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Gong, M. and Yang, Y. (2007). Real-time stereo matching using orthogonal reliability-based dynamic programming. *IEEE Trans. Image Processing*, 16(3):879–884.

Hannuna, S. (2007). *Quadruped Gait Detection in Low Quality Wildlife Video*. PhD thesis, University of Bristol. Supervisor-Neill Campbell.

Hartley, R. I. and Zisserman, A. (2001). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049.

Kanade, T. (1994). Development of a video-rate stereo machine. In *Image Understanding Workshop*, pages 549–9557.

Leonard, J. (2007). Challenges for autonomous mobile robots. pages 4–4.

Pauwels, K. and Hulle, M. M. V. (2004). Segmenting independently moving objects from egomotion flow fields. In *Proc. Early Cognitive Vision Workshop*.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA.

Rabe, C., Franke, U., and Gehrig, S. (2007). Fast detection of moving objects in complex scenarios. *Intelligent Vehicles Symposium, 2007 IEEE.*, pages 398–403.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42.

Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle.

Torr, P. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156.

Veksler, O. (2003). Extracting dense features for visual correspondence with graph cuts. In *IEEE CVPR*, pages –.

Wilson, J., Walker, B., Lindsay, J., and Dellaert, F. (2007). Swan: System for wearable audio navigation.

Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M., and Nister, D. (2006). Real-time global stereo matching using hierarchical belief propagation. In *BMVC*, pages –.

Yu, Q., Araújo, H., and Wang, H. (2005). A stereovision method for obstacle detection and tracking in non-flat urban environments. *Auton. Robots*, 19(2):141–157.

Yuan, C., Medioni, G. G., Kang, J., and Cohen, I. (2007). Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1627–1641.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vision*, 27(2):161–195.

Zitnick, C. and Kanade, T. (2000). A cooperative algorithm for stereo matching and occlusion detection. *IEEE Trans. PAMI*, 22(7):675–684.

Zitnick, C. and Kang, S. (2007). Stereo for image-based rendering using image over-segmentation. *Int. J. Comput. Vision*, 75:49–65.