3D Scene Segmentation and Object Tracking in Multiocular Image Sequences

Joachim Schmidt[1], Christian Wöhler[2], Lars Krüger[2], Tobias Gövert[1,2], and Christoph Hermes[1]

[1]University of Bielefeld, Technical Faculty, Applied Computer Science
P. O. Box 100131, D-33501 Bielefeld, Germany

[2]DaimlerChrysler AG, Group Research, Machine Perception
P. O. Box 2360, D-89013 Ulm, Germany

Abstract. In this contribution we describe a vision-based system for the 3D detection and tracking of moving persons and objects in complex scenes. A 3D point cloud of the scene is extracted by a combined stereo technique consisting of a correlation-based block-matching approach and a spacetime stereo approach based on spatio-temporally local intensity modelling, resulting in a 3D point cloud attributed with motion information. For localising persons and objects in the scene the point cloud is segmented into clusters by applying a hierarchical clustering algorithm, using velocity information as an additional discrimination criterion. Initial object hypotheses are obtained by partitioning the observed scene with cylinders, including the tracking results of the previous frame. Multidimensional unconstrained nonlinear minimisation is then applied to refine the initial object hypotheses, such that neighbouring clusters with similar velocity vectors are grouped to form a compact object. A particle filter is applied to select hypotheses which generate consistent trajectories. The described system is evaluated based on real-world sequences acquired in an industrial production environment and from a tabletop scene, using manually obtained ground truth data. We find that even in the presence of moving objects closely neighbouring the person, all objects are detected and tracked in a robust and stable manner. The average tracking accuracy is of the order of several percent of the distance to the scene.

1 Introduction

Three-dimensional (3D) vision plays an important role in human perception, especially for the recognition of motion and the ability to track objects over time. It is advantageous for vision systems to make use of this very basic information in order to perceive and interprete the environment.

An efficient approach to obtain 3D information about the scene without the need to place markers or to constrain the appearance of objects is stereo vision. A classical stereo vision algorithm is the block-matching approach [1–4]. At each pixel of, say, the left image, a rectangular window is centered on the position

of that pixel. The algorithm computes the disparity value for that pixel by determining in the right image the window of identical size on the same epipolar line for which a similarity measure, e.g. the sum of squared differences (SSD), obtains an optimum. A stereo vision algorithm that constructs dense depth maps uses Dynamic Programming [5]. It makes use of the ordering constraint, which requires that for opaque surfaces, the order of neighbouring correspondences on two corresponding epipolar lines is always preserved. Both previously described stereo vision approaches process each epipolar line independently. In contrast, the graph cut or maximum flow method optimises the solution globally. Instead of the ordering constraint, a more general local coherence constraint is assumed which claims that disparities tend to be locally similar. The correspondence problem can then be formulated as a maximum-flow problem in a graph [6]. Real-time stereo vision systems like those presented in [2] or [4] rely on the principle of establishing correspondence e.g. by computation of SSD. For speed-up, making use of several resolution levels [2] often turns out to be useful, as well as heuristics such as suppression of uniform image regions by employing an interest operator and checking for left-right consistency. The approach described in [2] is extended in [3] by tracking 3D points individually in a six-dimensional position-velocity space, thus extracting motion information during an additional processing step after correspondence analysis. While virtually all classical stereo vision approaches do not make use of image sequences, recent work [7] describes a block-matching spacetime stereo vision scheme which relies on pairs of image sequences rather than just pairs of images.

Depending on the number of available cameras, different methods for localising people in a scene can be applied. Using a single camera, a pedestrian is detected based on the appearance of a human [8] and it is possible to automatically initialise a tracker and roughly estimate the pose, representing body segments as 2D image patches [9]. If 3D data are available, e.g. from wide baseline stereo, localisation and tracking can be performed by mean shift clustering of the point cloud [10]. This system, however, is sensitive to noisy 3D data. For close-up observation of humans, a detailed articulated body model can be fitted to the generated 3D point cloud to estimate the pose of the human [11, 12]. Even human body tracking solely based on 2D image data is possible, but either strong models [13] or uncluttered background and multiple cameras [14] are needed.

2 3D scene analysis

In this section we will present our system for localising and tracking people and moving objects in an attributed 3D point cloud and discuss the underlying assumptions. We will employ a combination of a bottom-up approach for spatio-temporal scene segmentation and weak models for generation of object hypotheses in order to permit the detection of arbitrary objects. This approach allows task-dependent interpretation without incorporating strong models, which would restrain perception or require prior knowledge about the appearance or structure of the scene.

2.1 Correlation-based block-matching stereo algorithm

As a first stereo analysis algorithm, we employ the real-time block-matching approach described in [2]. We assume that the cameras are calibrated and the images are rectified to standard stereo geometry with epipolar lines parallel to the image rows [15]. For each interest pixel in the left image for which a sufficiently high intensity gradient is observed, a corresponding point is searched along the epipolar line in the right image. We use the sum of squared differences (SSD) as a similarity measure. A square region around the interest pixel in the left image is compared with regions on the corresponding epipolar line in the right image for all candidate disparities, resulting in an array of correlation coefficients. The disparity corresponding to the minimum SSD value is determined, and a parabola is fitted to the local neighbourhood of each maximum, yielding the disparity value at subpixel accuracy. Only well localised correspondences are considered for further processing.

2.2 Spacetime stereo based on local intensity modelling

Our second stereo approach exploits the spatio-temporal structure of the acquired sequence of image pairs. To the local spatio-temporal neighbourhood of each interest pixel a parameterised function $h(\boldsymbol{P}, u, v, t)$ is adapted, where u and v denote the pixel coordinates, t the time coordinate, and $\boldsymbol{P}$ the vector of function parameters.

Ideally, an object boundary is described by an abrupt intensity change. In real images, however, one does not observe such discontinuities since they are blurred by the point spread function of the optical system. Therefore, we model the intensity change at an object boundary by a "soft" sigmoid function like the hyperbolic tangent. As we cannot assume the image regions inside and outside the object to be of uniform intensity, we model the intensity distribution around an interest pixel by a combined sigmoid-polynomial approach:

$$h(\boldsymbol{P}, u, v, t) = p_1(v, t) \tanh\left[p_2(v, t)u + p_3(v, t)\right] + p_4(v, t). \tag{1}$$

The terms $p_1(v, t)$, $p_2(v, t)$, $p_3(v, t)$, and $p_4(v, t)$ denote polynomials in v and t. The polynomial $p_1(v, t)$ describes the amplitude and $p_2(v, t)$ the steepness of the sigmoid function, which both depend on the image row v, while $p_3(v, t)$ accounts for the row-dependent position of the model boundary. The value of $p_2(v, t)$ is closely related to the sign of the intensity gradient and to how well it is focused, where large values describe sharp edges and small values blurred edges. The polynomial $p_4(v, t)$ is a spatially variable offset which models local intensity variations across the object and in the background, e.g. allowing the model to adapt to cluttered background. All described properties are assumed to be time-dependent. Interest pixels for which no parametric model of adequate quality is obtained are rejected if the residual of the fit exceeds a given threshold.

The parametric model according to Eq. (1) in its general form requires that a nonlinear least-mean-squares optimisation procedure is applied to each interest

pixel, which may lead to a prohibitively high computational cost of the method. Is is possible, however, to transform the nonlinear optimisation problem into a linear one by assuming that (i) the offset $p_4(v,t)$ is proportional to the average pixel intensity $\bar{I}$ of the spatio-temporal neighbourhood of the interest pixel, i.e. $p_4(v,t) = w\bar{I}$, and (ii) the amplitude $p_1(v,t)$ of the sigmoid is proportional to the standard deviation σ_I of the pixel intensities in the spatio-temporal neighbourhood with $p_1(v,t) = k\sigma_I$. These simplifications yield the model equation

$$p_2(v,t)u + p_3(v,t) = \operatorname{artanh}\left[\frac{I(u,v,t) - w\bar{I}}{k\sigma_I}\right] \equiv \tilde{I}(u,v,t), \tag{2}$$

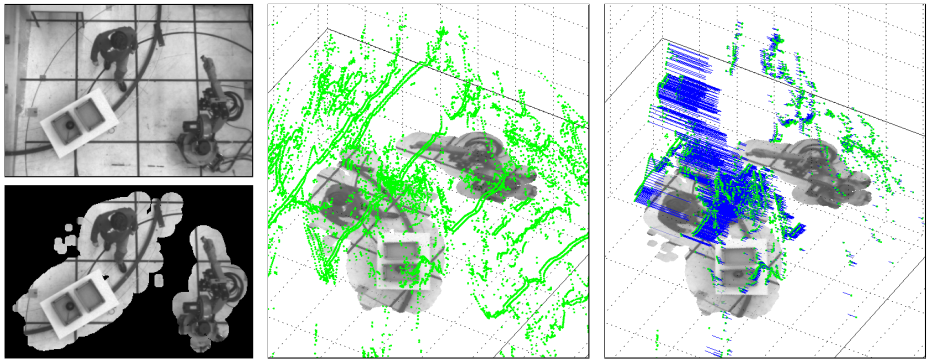
where the model parameters, i.e. the coefficients of the polynomials $p_2(v,t)$ and $p_3(v,t)$, can be determined by a linear fit to the transformed image data $\tilde{I}(u,v,t)$. Pixels with $\left|\left[I(u,v,t) - w\bar{I}\right] / \left[k\sigma_I\right]\right| > \theta$ are excluded from the fit, where θ is a user-defined threshold with $\theta < 1$, since arguments of the artanh function close to 1 would lead to a strong amplification of noise in the original pixel intensities $I(u,v,t)$. The factors k and w are further user-defined parameters of the algorithm.

The intensity gradient obtains its maximum value in horizontal direction at the root $u_e(v,t) = -p_3(v,t)/p_2(v,t)$ of the hyperbolic tangent. The horizontal position of the intensity gradient at the current time step for the epipolar line on which the interest pixel is located is given by the value $u_e(v_c,t_c)$, where the index c denotes the centre of the local neighbourhood of the interest pixel. The direction δ of the intensity gradient is given by $\delta = \partial u_e/\partial v$. The velocity μ of the intensity gradient along the epipolar line corresponds to the temporal derivative $\mu = \partial u_e/\partial t$ of the location of the epipolar transection. Both derivatives are computed at v_c and t_c.

For correspondence analysis, the SSD similarity measure is adapted to our algorithm by comparing the fitted functions $h(\boldsymbol{P}_l,u,v,t)$ and $h(\boldsymbol{P}_r,u,v,t)$ rather than the pixel intensities themselves, where the indices l and r denote the left and the right image, respectively:

$$S = \int \left[h(\boldsymbol{P}_l, u - u_e^l(v_c,t_c), v, t) - h(\boldsymbol{P}_r, u - u_e^r(v_c,t_c), v, t)\right]^2 du\ dv\ dt, \tag{3}$$

where u, v, and t traverse the local spatio-temporal neighbourhood of the left and the right interest pixel, respectively. Once a correspondence between two interest pixels on the same epipolar line has been established by searching for the best similarity measure, the disparity d corresponds to $d = \left[u_i^l + u_e^l(v_c,t_c)\right] - \left[u_i^r + u_e^r(v_c,t_c)\right]$ with u_i^l and u_i^r as the integer-valued horizontal pixel coordinates of the left and the right interest pixel, respectively. Given the optical and geometrical parameters of the camera system, the velocity components parallel to the epipolar lines and along the depth axis can be computed directly in metres per second from the values $\bar{\mu} = \left(\mu^l + \mu^r\right)/2$ and $\partial d/\partial t = \mu^l - \mu^r$.



top: (a), bottom: (b) (c) (d)

Fig. 1. (a) Original image (left camera), (b) background subtracted image, (c) full correspondence stereo point cloud, (d) reduced motion-attributed point cloud.

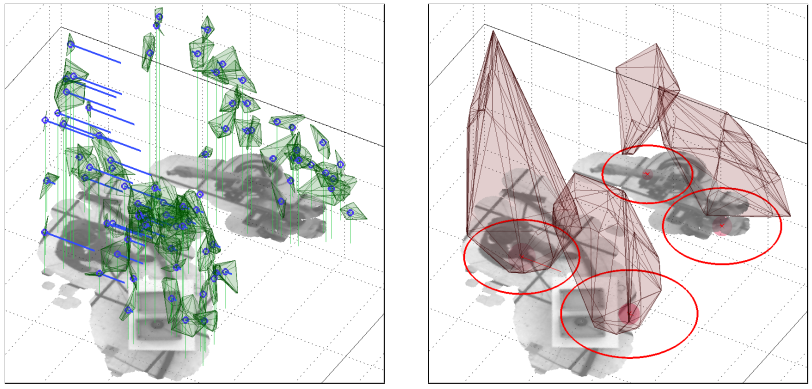
2.3 Motion-attributed point cloud

Both described stereo techniques generate 3D points based on edges in the image, especially object boundaries. Due to the local approach they are independent of the object appearance. While correlation stereo has the advantage of higher spatial accuracy and is capable of generating more point correspondences, spacetime stereo provides a velocity value for each stereo point. However, it generates a smaller number of points and is spatially less accurate, since not all edges are necessarily well described by the model defined in Eq. (1). Taking into account these properties of the algorithms, the results are merged into a single motion-attributed 3D point cloud. For each extracted 3D point c_k an average velocity $\bar{v}(c_k)$ is calculated, using all spacetime points s_j, $j \in (1, \ldots, J)$ in an ellipsoid neighbourhood defined by $\delta_S(s_j, c_k) < 1$ around c_k. To take into account the spatial uncertainty in depth direction of the spacetime data, $\delta_S(s_j, c_k)$ defines a Mahalanobis distance whose correlation matrix Σ contains an entry $\Sigma_z \neq 1$ for the depth coordinate which can be derived from the recorded data.

$$\bar{v}(c_k) = \frac{\rho}{J} \sum_{j=1}^{J} v(s_j) \; \forall \; s_j : \delta_S(s_j, c_k) < 1 \tag{4}$$

The factor ρ denotes the relative scaling of the velocities with respect to the spatial coordinates. It is adapted empirically depending on the speed of the observed objects. This results in a 4D point cloud, where each 3D point is attributed with an additional 1D velocity component parallel to the epipolar lines, see Fig. 1(d).

A reference image of the observed scene is used to reduce the amount of data to be processed by masking out 3D points that emerge from static parts of the scene, as shown in Fig. 1(a,b). Furthermore, only points within a given interval above the ground plane are used, as we intend to localise objects and humans and thus always assume a maximum height for objects above the ground.



(a) (b)

Fig. 2. (a) Over-segmentation and cluster velocities, (b) objects with convex hull.

2.4 Over-segmentation for motion-attributed clusters

To simplify the scene representation, we apply a hierarchical clustering algorithm, recognising small contiguous regions in the cloud, based on features like spatial proximity or homogeneity of the velocities. This procedure deliberately over-segments the scene, generating motion-attributed clusters. By incorporating velocity information for clustering, we expect an improvement in segmentation at these early stages of the algorithm, without needing strong models to ensure separation of neighbouring objects. For clustering, we apply the complete linkage algorithm [16], also called furthest neighbour, to describe the distance between two clusters. The resulting hierarchical tree is partitioned by selecting a clustering threshold and addressing each subtree as an individual cluster, see Fig. 2(a). The criterion for selecting the threshold is the increase in distance between two adjacent nodes in the tree, for which a maximum allowed value is determined empirically. For each resulting cluster l, the weight $w(l)$ is set according to the number of points P belonging to l: $w(l) = \sqrt{P}$. The square root is used to constrain the weight for clusters consisting of many points. For each cluster the mean velocity of all points belonging to it is determined.

2.5 Generation and tracking of object hypotheses

From here on, persons and objects can be represented as a collection of clusters of similar velocity within an upright cylinder of variable radius. An object hypothesis $R(a)$ is represented by a four-dimensional parameter vector $a = [x\ y\ v\ r]$, with x and y being the centre position of the cylinder on the ground plane, v denoting the velocity of the object and r the radius. This weak model is suitable for persons and most encountered objects.

To extract the correct object positions, we utilise a combination of parameter optimisation and tracking. We first generate a number of initial hypotheses, optimise the location in parameter space, and then utilise the tracking algorithm to

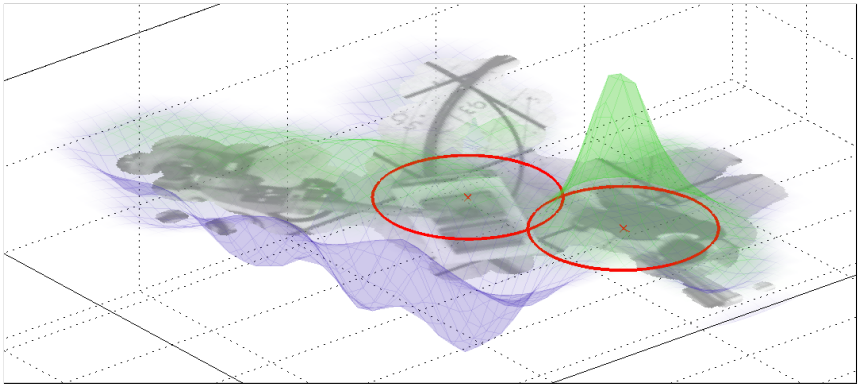


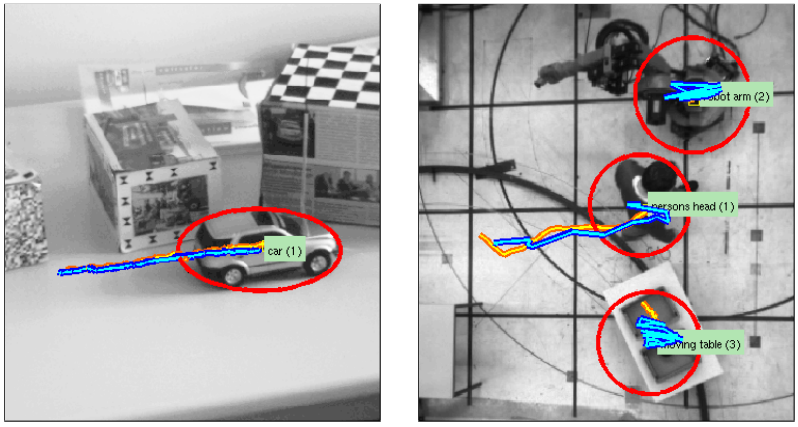
Fig. 3. Error function plot for minimisation, showing the XY error surface for $v = 0.26$ m s^{-1}, $r = 0.53$ m (blue), and $v = -0.79$ m s^{-1}, $r = 0.53$ m (green, values mirrored for clearer display).

select hypotheses which form consistent trajectories. Initial object hypotheses are created at each time step by partitioning the observed scene with cylinders and by including the tracking results from the previous frame, respectively. Multidimensional unconstrained nonlinear minimisation [17] is applied to refine the position and size of the cylinders in the scene, so that as many as possible neighbouring clusters with similar velocity values can be grouped together to form compact objects, as shown in Fig. 2(b). An error function $f(a)$ used for optimisation denotes the quality of the grouping process for a given hypothesis. Each hypothesis is weighted based on the relative position, relative velocity, and weight of all clusters l within the cylinder $R(a)$ using Gaussian kernels:

$$f(a) = f_r(a) \sum_{l \in R(a)} w(l) f_d(l,a) f_v(l,a) \tag{5}$$

with $f_r(a) = \exp\left(-\frac{r(a)^2}{2H_{r,\min}^2}\right) - \exp\left(-\frac{r(a)^2}{2H_{r,\max}^2}\right)$ keeping the radius in a realistic range, $f_d(l) = \exp\left(-\frac{[s(l)-s(a)]^2}{2H_d^2}\right)$ reducing the importance of clusters further away from the cylinder centre, and $f_v(l,a) = \exp\left(-\frac{[v(l)-v(a)]^2}{2H_v^2}\right)$ masking out clusters having differing velocities. The functions $r(a)$, $s(a)$, and $v(a)$ extract the radius, the 2D position on the ground plane and the velocity of the hypothesis a respectively. The kernel widths H are determined empirically. Fig. 3 shows the error function from Eq. (5), parameterised for opposing velocities. Local minima are centred on top of the objects of interest.

After optimisation, hypotheses with identical parameterisation are merged and those without any clusters within $R(a)$ are removed. The remaining hypotheses are tracked over time using a particle filter, keeping only object hypotheses forming a consistent trajectory.



(a) (b)

Fig. 4. Trajectories of tracked objects (blue) with annotated ground truth (yellow): (a) tabletop and (b) industrial scene.

3 Evaluation

For evaluation of our system, we used two types of real-world image sequences recorded with a PointGrey Digiclops multiple CCD camera system with an image size of 640×480 pixels, a pixel size of 7.4 μm, and a focal length of 4 mm. The stereo baseline corresponds to 100 mm. One sequence displays a moving toy car on a table driving past static objects, the other sequences an industrial working cell with a human worker, a robot, and a moving platform. The distance to the scene is 1.47 m for the tabletop sequence and 5.65 m for the industrial sequences.

We empirically found for the correlation matrix element Σ_z in Eq. (4) the value $\Sigma_z = 0.292$, regarding a set of 3D points obtained with the spacetime stereo algorithm and belonging to a plane scene part, while $\Sigma_x = \Sigma_y = 1$ are equally scaled. The velocity scaling factor is set to $\rho = 380$ s, where the velocity is expressed in metres per second and the spatial coordinates in metres. The kernel widths for Eq. (5) are chosen as $H_{r,\max} = 1.88$ m, $H_{r,\min} = 0.135$ m, $H_d = 0.113$ m, and $H_v = 0.188$ m for the industrial scenes. While the object sizes as well as the overall size of the scenes are far different, the kernel widths merely need to be scaled by an empirical uniform factor, such that the relative parameter values remain constant. The value of ρ depends on the typical velocities encountered in the scene. Hence, we set for the tabletop scene $H_{r,\max} = 4.14$ m, $H_{r,\min} = 0.297$ m, $H_d = 0.249$ m, $H_v = 0.414$ m, and $\rho = 3200$ s.

For each sequence, ground truth was generated manually by marking the center of the objects of interest in each frame, e. g. the head of the person or the center of the car, and transforming them into 3D coordinates using the known geometry of the scene and the objects, e.g. the tallness of the person and the position of the ground plane. The trajectories of the tracked objects are compared to the ground truth based on the corresponding value of the root mean square error (RMSE). The results in Table 1 show that objects can be tracked in

seq.	# pic.	object	with velocity RMSE	with velocity % tracked	without velocity RMSE	without velocity % tracked
tabletop	95	car	2.45	100.0	2.59	100.0
industry1	69	person	26.5	100.0	38.3	84.8
		table	60.3	100.0	21.8	69.7
		robot	87.8	95.5	111.8	98.5
industry2	79	person	42.7	100.0	31.8	94.8
		table	43.5	100.0	27.5	100.0
		robot	12.1	98.7	17.7	96.1
industry3	24	person	19.6	100.0	14.7	100.0
		table	24.9	100.0	22.5	90.9
		robot	17.1	100.0	29.3	100.0
industry4	39	person	24.7	75.7	35.2	89.2
		table	27.0	100.0	24.5	97.3
		robot	9.1	100.0	20.0	97.3
industry5	24	person	20.8	90.9	25.4	81.8
		table	21.9	100.0	32.9	100.0
		robot	8.6	77.3	33.1	100.0

Table 1. Tracking results compared to ground truth. RMSE is given in centimetres.

a stable manner at reasonable accuracy. Using velocity as an additional feature yields a more accurate localisation result for 10 of 16 detected objects, and detection is usually possible in a larger fraction of the frames. For four other objects the RMSE but at the same time also the detection rate is lower when velocity information is neglected. The system is designed to segment the point cloud into clusters of differing velocity. As a consequence, the proposed system works best for objects with homogeneous velocity. For example, we observed that for a walking person moving the arms backwards the object hypothesis does not contain the arms. As it is illustrated by the trajectories in Fig. 4, the system is able to track objects and persons in a top-view surveillance setup as well as in a side-view setup.

4 Conclusion and outlook

In this paper we have described a vision-based system for 3D detection and tracking of moving persons and objects in complex scenes. By combining correlation and spacetime stereo results, robust clustering of neighbouring objects in a motion-attributed 3D point cloud can be achieved. Objects and persons in the scene are localised and tracked incorporating velocity information. Our evaluation verifies the applicability of the system to different scenarios and the advantage of using velocity information as an additional clustering criterion. The localisation accuracy amounts to a few centimetres for the tabletop scene and is of the order 0.1–0.3 m for the industrial scenes. Only the velocity component along epipolar lines is taken into account, since no significant radial motion occurs in the regarded scenes. Perpendicular motion components could be integrated using a second camera pair. Future work will address the analysis of our

system with respect to segmentation quality in the presence of noisy velocity information and its applicability in the field of model-based body pose tracking.

References

1. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge, Massachusetts (1993)
2. Franke, U., Joos, A.: Real-time stereo vision for urban traffic scene understanding. In: Conf. on Intelligent Vehicles, Detroit, IEEE (2000)
3. Franke, U., Rabe, C., Badino, H., Gehrig, S.K. Lecture Notes in Computer Science 3663. In: 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. Pattern recognition. proc. 27th dagm symposium, vienna, austria edn. Springer-Verlag Berlin Heidelberg (2005) 176–183
4. Hirschmueller, H.: Improvements in real-time correlation-based stereo vision. In: Int. Conf. on Computer Vision and Pattern Recognition, Stereo Workshop, Hawaii (2001)
5. Cox, I., Hingorani, S., Rao, S.: A maximum likelihood stereo algorithm. Computer Vision and Image Understanding vol.63(3) (1996)
6. Roy, S., Cox, I.: A maximum-flow formulation of the n-camera stereo correspondence problem. In: Int. Conf. on Computer Vision, Bombay (1998) 492–499
7. Davis, J., Nehab, D., Ramamoorthi, R., Rusinkiewicz, S.: Spacetime stereo: A unifying framework for depth from triangulation. IEEE Trans. Pattern Analysis and Machine Intelligence vol.27(2) (2005)
8. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. Technical report, Mitsubishi Electric Research Lab Technical Report TR-2003-90 (2003)
9. Ramanan, D., Forsyth, D.: Finding and tracking people from the bottom up. In: Conference on Computer Vision and Pattern Recognition - CVPR. (2003)
10. Keck, M., Davis, J., Tyagi, A.: Tracking mean shift clustered point clouds for 3d surveillance. In: ACM Mulimedia Workshop on Video Surveillance and Sensor Networks, Santa Barbara, California, USA, VSSN (2006) 187–194
11. Knoop, S., Vacek, S., Dillmann, R.: Modeling joint constraints for an articulated 3d human body model with artificial correspondences in icp. In: Proceedings of the International Conference on Humanoid Robots, Tsukuba, Japan (2005)
12. Ziegler, J., Nickel, K., Stiefelhagen, R.: Tracking of the articulated upper body on multi-view stereo image sequences. In: Conference on Computer Vision and Pattern Recognition - CVPR, New York, USA, IEEE Computer Society (2006)
13. Schmidt, J., Kwolek, B., Fritsch, J.: Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images. In: Proc. of Automatic Face and Gesture Recognition, Southampton, UK, IEEE (2006) 567–572
14. Rosenhahn, B., Kersting, U., Smith, A., Gurney, J., Brox, T., Klette, R.: A system for marker-less human motion estimation. In Kropatsch, W., Sablatnig, R., Hanbury, A., eds.: Pattern recognition : 27th DAGM Symposium. Volume 3663 of Lecture Notes in Computer Science., Vienna, Austria, Springer (2005) 230–237
15. Krüger, L., Wöhler, C., Würz-Wessel, A., Stein, F.: In-factory calibration of multiocular camera systems. In: SPIE Photonics Europe (Optical Metrology in Production Engineering), Strasbourg (2004) 126–137
16. Berthold, M., Hand, D.J., eds.: Intelligent Data Analysis. 2nd edn. Springer (2003)
17. Nelder, J.A., Mead, R.: A simplex method for function minimization. Computer Journal vol.7 (1965) 308–313