

3-D-Skeleton-Based Head Detection and Tracking Using Range Images

Pandu Rangarao Devarakota, *Student Member, IEEE*, Marta Castillo-Franco, Romuald Ginhoux, *Member, IEEE*, Bruno Mirbach, Serge Kater, and Björn Ottersten, *Fellow, IEEE*

Abstract—Vision-based 3-D head detection and tracking systems have been studied in several applications like video surveillance, face-detection systems, and occupant posture analysis. In this paper, we present the development of a topology-based framework using a 3-D skeletal model for the robust detection and tracking of a vehicle occupant's head position from low-resolution range image data for a passive safety system. Unlike previous approaches to head detection, the proposed approach explores the topology information of a scene to detect the position of the head. Among the different available topology representations, the Reeb graph technique is chosen and is adapted to low-resolution 3-D range images. Invariance of the graph under rotations is achieved by using a Morse radial distance function. To cope with the particular challenges such as the noise and the large variations in the density of the data, a voxel neighborhood connectivity notion is proposed. A multiple-hypothesis tracker (MHT) with nearest-neighbor data association and Kalman filter prediction is applied on the endpoints of the Reeb graph to select and filter the correct head candidate out of Reeb graph endpoints. A systematic evaluation of the head detection framework is carried out on full-scale experimental 3-D range images and compared with the ground truth. It is shown that the Reeb graph topology algorithm developed herein allows the correct detection of the head of the occupant with only two head candidates as input to the MHT. Results of the experiments demonstrate that the proposed framework is robust under the large variations of the scene. The processing requirements of the proposed approach are discussed. It is shown that the number of operations is rather low and that real-time processing requirements can be met with the proposed method.

Index Terms—Airbags, automotive safety, multiple-hypothesis tracking (MHT), Reeb graph, 3-D head detection and tracking, 3-D skeleton.

Manuscript received December 24, 2007; revised June 13, 2008, September 24, 2008, and January 6, 2009. First published April 10, 2009; current version published October 2, 2009. This work was supported in part by IEE S. A., Luxembourg, and in part by the Luxembourg International Advanced Studies in Information Technology, Luxembourg. The review of this paper was coordinated by Dr. M. S. Ahmed.

P. R. Devarakota is with the IEE S. A., 5326, Contern, Luxembourg, and also with the Signal Processing Group, School of Electrical Engineering, Royal Institute of Technology, 100 44, Stockholm, Sweden (e-mail: pandu.devarakota@ee.kth.se).

M. Castillo-Franco, R. Ginhoux, B. Mirbach, and S. Kater are with the IEE S. A., 5326, Contern, Luxembourg (e-mail: marta.castillo@iee.lu; romuald.ginhoux@iee.lu; bruno.mirbach@iee.lu; serge.kater@iee.lu).

B. Ottersten is with the Signal Processing Group, School of Electrical Engineering, Royal Institute of Technology, 100 44, Stockholm, Sweden, and also with securityandtrust.lu, University of Luxembourg, 1359 Luxembourg, Luxembourg (e-mail: bjorn.ottersten@ee.kth.se).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVT.2009.2020595

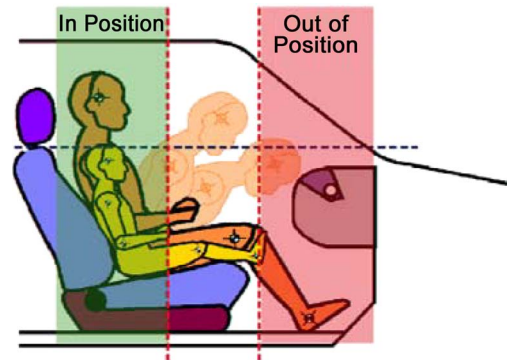


Fig. 1. *In-position* and *out-of-position* zones of the passenger seat. By monitoring the passenger's head, the 3-D vision system detects the critical passenger position. The airbag should not be triggered if the passenger's head is located in the *out-of-position* area at the instant of the crash.

I. INTRODUCTION

VISION-BASED 3-D head detection and tracking systems have been studied in many applications, for example, video surveillance, face-detection systems, and occupant posture analysis systems and tracking [1]–[4]. In these applications, the challenge is mainly to cope with the large variations of the scene, viewing perspectives, and illumination conditions and to provide real-time performance. In the field of automotive safety, a 3-D vision system based on a novel range sensor was recently investigated to solve the occupant classification problem for the optimal deployment of airbags. [5]. This range sensor is based on the time-of-flight principle with an active illumination source using near-infrared light. Compared with the other 3-D vision systems proposed in the past for the same application (see, for example, [3], [6], and [7]), this system has an advantage of directly providing a dense range image that is independent of illumination conditions and object textures.

This system can also be used to detect the size, position, and posture of the occupant. In particular, the head position of the occupant gives valuable input to the airbag's control unit to deploy the airbag at an optimal level based on the occupant position, that is, to provide optimal protection while reducing the risk of injury due to an airbag deployment.

For this purpose, the passenger's head should be localized with respect to the vehicle coordinate system. In case of a so-called *out-of-position* situation where the head is too close to the airbag module, the airbag should be depowered or should not be triggered at all (see Fig. 1).

Robust head detection in this situation is a challenging problem due to the constraints of the application. Face-recognition

methods for head detection are not applicable as the face may not always be visible since the camera is mounted in the overhead module of the car, giving a side view of the passenger rather than a frontal view. Moreover, a low-resolution camera like the one studied in [5] does not resolve the details needed for face detection. Typically, the head detection is therefore based on analyzing the local shape of headlike objects in the data. The typical model of a head with an elliptical shape (e.g., finding ellipselike shapes in the scene in case of monocular cameras or finding the ellipsoidlike 3-D shapes in the scene in case of 3-D cameras) is often considered for this purpose [3], [8]–[10]. The classification is then based on the shape features extracted from each headlike shape detected in the data. However, the robustness of such head detection methods can be an issue due to the following reasons. On one hand, the shape of a human head varies a lot due to hairstyles, hats, etc. On the other hand, there are many regions in a scene, such as the shoulder and the elbow of the occupant, that locally have the shape of the head model. In addition, there may be other objects that occupants can carry, for example, a balloon, that fit even better to the head model than the occupant's head itself. Moreover, the number of candidates detected by a geometrical approach cannot be limited *a priori*.

The only chance to overcome these situations where possible false detection may occur is to take into account not only the local shape features but also the topology of the whole scene. The basic idea is that an occupant of a vehicle has a typical topology that is independent of the occupant size and other variations like hairstyles that change the geometry of the occupant. The topology can mathematically be represented in terms of a skeleton or a graph. The important parts of the scene can be represented by the branches of the skeleton. For instance, the skeleton of the scene where the passenger seat is normally occupied by a person is represented by three branches: one corresponding to the backrest, the second corresponding to the upper torso of the person, and the third corresponding to the person's legs. This way, the possible head candidates can be identified as the endpoints of the skeleton, and their positions can be determined from the coordinates of the data points associated to the endpoints. The topology information is very attractive since it does not require any geometrical model to be assumed or any local feature analysis. This paper presents a general topology-based framework for low-resolution range images that offers a systematic way to detect the vehicle occupant's head.

A. Topology Coding of 3-D Objects

The topology of an object represented in a skeletal structure has proven to be very useful in various applications of computer vision, for instance, 3-D indexing [11], classification [12], and 3-D object reconstruction [13]. The Reeb graph is one method for representing the topology as a skeleton that shows how different parts of an object are connected to each other [14]–[16]. The goal of this paper is not to show that the Reeb graph method is superior to other alternative skeletonization methods but to show how topology information is useful to localize the position of the occupant's head in this automotive safety

application. Topological representations other than the Reeb graph can also be used under similar conditions.

Reeb graphs are based on the Morse theory [14], which says that the change of topology of a manifold can be detected by identifying the critical points of a smooth function defined on the manifold. A point is a critical point if and only if the derivative of the smooth function with respect to the local parameterization of the manifold vanishes. The Reeb graph graphically shows the configuration of all identified critical points and their topological connectivity. It is initially proposed to represent the skeleton of a compact manifold, and it is therefore useful to understand the intrinsic topological structure of the shape of a manifold [15]. Using the knowledge of critical points, the way a manifold is embedded in a 3-D space can be coded in terms of the Reeb graph.

One of the main concerns in the Reeb graph extraction is its invariance. The extracted Reeb graph should be invariant under translation, rotation, and scaling based on the application. For example, in object recognition applications, one prefers to have a Reeb graph invariant under different realistic variations that may occur on the object. By defining a proper Morse function, this can be achieved. There are several functions studied in the literature. Each function has different properties and suitability to the type of application. The height function introduced in [13] is suitable for models spread along the vertical axis, such as a human being representation. The function returns the value of the z -coordinate if it is defined on a 2-D manifold. The height function is easy to compute, but it is variant to object rotations. In [17], a Morse distance function was proposed, and it was shown therein that it is invariant to translation, scale, and rotation. Due to its simplicity and invariant properties, we consider the distance function in our method as a Morse function.

The traditional Reeb graph approach requires a dense 3-D representation of the closed object surface, e.g., a triangulated mesh representation. However, this is not the case for the 3-D representation extracted from a range image. There, one obtains a cloud of 3-D points representing the part of an object surface that is visible from the camera point of view. This kind of data does not have the same connectivity as dense volumetric data and can rather be seen as 2.5-D data. A second property of these data is that the density of data points is not constant but varies with the distance of the object surface points to the camera, as is the case for a wide-angle camera surveilling the passenger seat inside a car. Moreover, data drawn from real images are subject to noise in the range measurements. Therefore, the traditional methods for extracting the Reeb graph must be refined and adapted to the current application and specific characteristics of the 3-D data at hand.

B. Contributions and Organization of the Paper

The contributions of this paper can be summarized as follows:

- 1) detection of the occupant's 3-D head position using a novel 3-D range camera technology;
- 2) novel topology-based framework for head detection and tracking;

- 3) extraction of a Reeb graph from range image data, thereby handling the variations in the data density by establishing a voxel connectivity;
- 4) reduction of the number of head candidates to a maximum of two by combining the topological approach with application-specific geometrical constraints;
- 5) evaluation of a multiple-hypothesis tracking (MHT) algorithm to locate the correct head location for each frame in a sequence.

As a first task, we propose here an adaptation of the Reeb graph method to our 2.5-D data, which is motivated by the work of Xiao *et al.* [18], who proposed a discrete Reeb graph (DRG) approach to 3-D point-cloud data obtained by a full scan of the object to segment different parts of the human body. They explored the connectivity based on a pairwise distance between the 3-D data points. For our data, we propose to rather use the neighborhood connectivity of voxels for building a Reeb graph. The benefit of using a voxel representation of the 2.5-D data is that by choosing an appropriate activation level for each voxel, depending on its distance to the camera, one can cope with the large variation of the density of data and eliminate outliers that can occur due to noise in the data. Moreover, voxel connectivity is computationally more efficient than calculating pairwise distance, which is important for the development of an automotive application that is also subject to cost constraints.

For the tracking task, to track the multiple head candidates given by the Reeb graph, the MHT algorithm is considered [19]. The aim of the MHT algorithm is to generate multiple trajectories for the instantaneously detected head candidates and to select out of them the most likely head position in every frame of the sequence. It is, however, known that one of the key issues in the MHT algorithm is the combinatorial explosion of hypothesis over the time. As a result of the above fact, the MHT algorithm is computationally exponential in both time and memory. One of the strengths of the proposed topological approach is that it yields an *a priori* limited number of candidates, thus limiting the potential combinatorial explosion of hypothesis in the MHT. To solve the candidate association ambiguity, a nearest-neighbor data association algorithm with a Mahalanobis distance test is considered. Based on the association matrix, tracks are updated using a linear Kalman filter with a nearly constant velocity model. Accelerations of vehicle occupants are thereby treated as model noise. For initially selecting the right head candidate that is close to the true head position, a simple rule has been derived based on the evaluation of the topology and geometrical constraints of the situation in the car.

This paper is organized as follows. We first briefly describe the experimental setup with a range data acquisition inside a vehicle in Section II. The proposed modified Reeb graph concept for the head detection is described in detail in Section III. Section IV describes the MHT algorithm proposed to handle multiple head candidates given by the Reeb graph. The experimental evaluation of the proposed method and results are presented in Section V. A detailed discussion on the processing requirements of the proposed approach is presented in Section VI. This paper finishes with conclusions in Section VII.

II. DATA ACQUISITION AND EXPERIMENTAL SETUP

A novel 3-D low-resolution range camera that is based on the time-of-flight principle is used [20]. The advantage of this camera is that it directly provides a dense range image, independent of the ambient illumination conditions and object textures. The scene is broadly illuminated by a modulated infrared light beam, at a frequency of 20 MHz; then, the modulated beam is reflected by an object and is detected by the receiver. Due to the time of flight of light to and from the target, the detected beam has a phase shift compared to the phase of the modulation signal in the illumination. This phase difference $\Delta\phi$ can be calculated by sampling the received light intensities $a_0, \dots, a_3$ at four temporal positions corresponding to a phase shift of 0, $\pi/2$, π , and $3\pi/4$, respectively. The amplitude and phase shift are then given by

$$A = \frac{\sqrt{(a_3 - a_1)^2 + (a_0 - a_2)^2}}{2}$$

$$\Delta\phi = \text{atan} \left[\frac{a_3 - a_1}{a_0 - a_2} \right]. \quad (1)$$

The measured phase delay can directly be converted into the distance between the target and the camera, which is denoted as L , according to

$$L = L_{\max} \frac{\Delta\phi}{2\pi} \quad (2)$$

where the parameter $L_{\max}$ is called the nonambiguity range of the sensor ($L_{\max} = c/2f_{\text{mod}}$, with f_{mod} being the modulation frequency of illumination) [21]. For the sample used in this paper, the modulation frequency has been 20 MHz, yielding a nonambiguity range of 750 cm.

By independently performing this phase measurement but in parallel for each pixel of the image, a range image (also called a distance image) is acquired, i.e., a 2-D image where each pixel value represents the radial distance from the camera center to a point in the scene at which the corresponding pixel is looking.

The accuracy of the distance measurement is mainly determined by the amplitude A of the received light (1), which is dependent on the object remission coefficient and the distance of the object to the camera (see also [22]). Consequently, an amplitude threshold has been introduced, such that the standard deviation of the distance measurement noise is below 5 cm if A is above this threshold. For amplitudes below this threshold, the distance in (2) is not computed, and the corresponding pixel is marked with black. The resolution of the system, that is, the accuracy that can be obtained under ideal conditions, is around 1.0 cm.

The camera has been equipped with a wide-angle lens, providing a field of view of $120^\circ \times 90^\circ$ and has been installed in the overhead module of a car, as shown in Fig. 2. The lens maps the scene in the car onto an imager with 52×50 pixels where each pixel provides the radial distance of the camera lens to the point in the scene that is optically mapped onto the corresponding pixel. For the given field of view, this number of pixels provides a lateral resolution of 3 cm $\times$ 2 cm at a distance of 50 cm, which corresponds to the average distance of an



Fig. 2. Test vehicle equipped with a prototype camera mounted in the overhead module; the camera is able to survey the complete passenger seat.

occupant from the camera. The frame rate of the camera T_{cam} is 14 frames/s. The size of the camera is $13.9 \text{ cm} \times 4.2 \text{ cm} \times 6 \text{ cm}$. See [22] for more details on the camera technology.

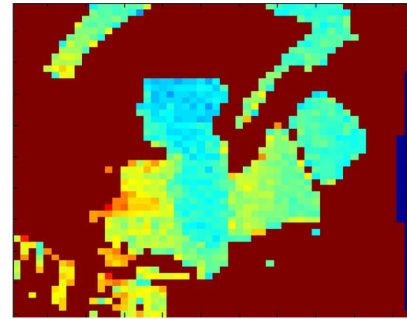
Fig. 3(a) shows an example of a range image in false-color representation, and Fig. 3(c) shows an intensity image of the same scene that was taken with a high-resolution 2-D camera to provide a reference. The segmentation of the occupant from the background can easily be realized by comparing the range image with a reference range image of the car interior. As the position of the headrest and seat may vary, they cannot be removed by this so-called background clipping and are therefore considered to be a part of the occupant scene instead. Fig. 3(b) shows the same image as in Fig. 3(a) after background clipping and filtering. Note that for the dark brown (black) pixel in Fig. 3(a), the corresponding distances are not computed as the respective amplitude of the received light for that pixel is smaller than a certain threshold value.

III. METHOD FOR HEAD DETECTION FROM THE REEB GRAPH

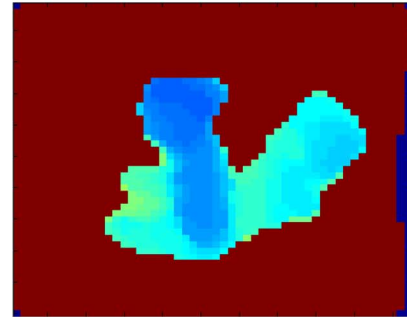
A. Reeb Graph Extraction on Range Data

In the past, the Reeb graph extraction has been studied for 2-D and 3-D images (see, for instance, [13], [15], and [17]). A study of the Reeb graph extraction from range data was reported in [11] and was used for the task of 3-D indexing. These studies are based on the assumption that 3-D objects that are modeled as explicit surfaces such as triangulated meshes or parametric surfaces are readily available at hand, which are otherwise time consuming to compute. In addition, the possible definitions of critical points generally suffer from instability since small perturbations of the vertex coordinates (in case of a triangulated mesh representation) may result in rather different configurations. In [12], an extended Reeb graph approach to a discrete surface is proposed, where the data are described with polygonal contours. These representations, however, are not ideal for potentially noisy and incomplete data such as 3-D points produced by a range sensor. Moreover, the direct application of the Reeb graph on range images is not possible as the data does not comply with the assumption of a smooth manifold.

The construction of a Reeb graph for dense range data obtained by a full scan around the object was proposed in [18]. The authors explored the use of Reeb graphs to segment human



(a)



(b)



(c)

Fig. 3. Example of a range image taken from a person on the passenger seat. (a) Raw range image in false-color representation. Blue (dark gray) represents the closest points to the camera, and red (light gray) presents the farthest points from the camera. (b) Same range image clipping the background and filtering. The remaining object shows the person and the vehicle seat. (c) High-resolution image of the same scene is shown.

body scans. The extracted Reeb graph was referred therein as DRG. The connectivity between any two points is defined based on a threshold on the value of the absolute difference between them. Our method is motivated by the approach in [18]; we have, however, chosen the concept of voxel neighborhood to establish connectivity in the graph. The voxel representation of the dense range data is particularly advantageous as the variations of density of the data can be taken into account by assigning to each voxel an activation level depending on the distance to the camera (see below). Moreover, the voxel representation allows the removal of outliers that may occur due to noise in the range measurement.

B. Voxel Neighborhood

For all experiments, the range image that is represented in Cartesian coordinates is used. That is, the range image is transformed to Cartesian coordinates in the vehicle coordinate

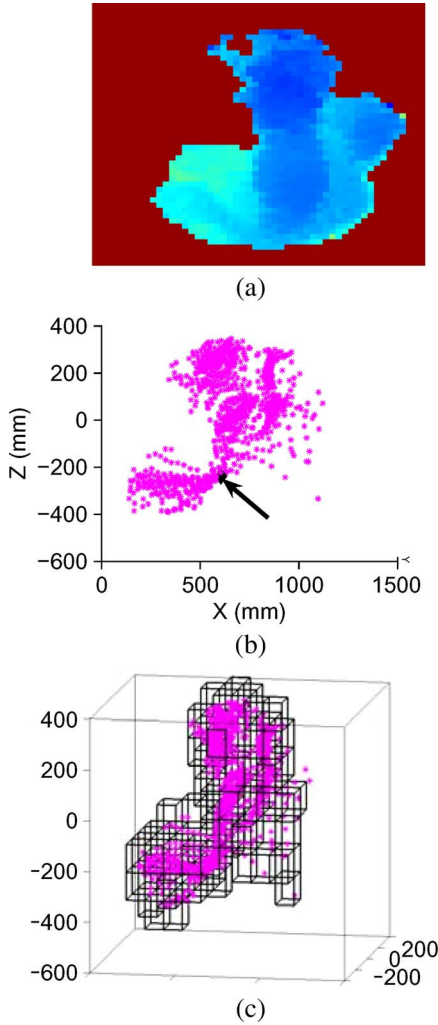


Fig. 4. (a) Example of a clipped range image. (b) Scatter plot of Cartesian coordinates extracted from the range image in the vehicle coordinate system (only (x, z) shown). (c) Corresponding voxel representation of the 3-D-scatter plot. The arrow in (b) indicates the starting point for the Reeb graph extraction algorithm.

system, where a triple (x, y, z) is assigned to each pixel, and this is realized as follows.

- On system initialization, a set of unit vectors is calculated from the internal and external camera parameters. Each pixel is associated with a unit vector pointing to the direction from where it receives light. These unit vectors are kept in memory during the runtime of the system.
- The transformation to vehicle coordinates is achieved by multiplying the range image point with the unit vectors and by adding a constant translation that represents the offset between the origin of the camera and the vehicle coordinate system.

Fig. 4(a) shows an example of a range image, and a scatter plot of the corresponding coordinate points is shown in Fig. 4(b).

The first step of our method consists of transforming these 3-D data points into a volume representation. For this purpose, 3-D grids of equally sized cube volume elements (voxels) are used. That is, the 3-D coordinate space is sampled using a regularly spaced array of voxel elements, and each voxel can

be seen as a box in the Cartesian space. The size of the voxel is thereby a parameter that is adapted to the data and to the application. That is, the voxel size is chosen to be larger than the minimum resolution of the camera but small enough to resolve the details of the occupant scene topology we are interested in. We have chosen an equal voxel size of 100 mm, which corresponds to the average radius of a human's head. Fig. 4(c) shows the voxel representation of the scene shown in Fig. 4(b) as an example.

Each pixel in the image is uniquely assigned to one voxel, but each voxel can contain several pixels. A voxel is said to be activated only if it contains a number of pixels larger than a so-called activation level. As the density of 3-D data locally depends on the distance of the object to the camera, we have chosen the activation level as C/d^2 , where d^2 is the mean square distance of all pixels the voxel element contains, and the constant C is chosen such that the activation level varies from 1 pixel at the largest possible distance to 10 pixels at the mean distance of the camera to the occupant. This way, a single outlier will not lead to an activated voxel.

The extraction of the Reeb graph is now based on the notion of connectivity described in the following definitions.

Definition 1 (Concept of Voxel Neighborhood): Two voxel elements are said to be 6-adjacent if they share a face, 18-adjacent if they share a face or an edge, and 26-adjacent if they share a face, an edge, or a vertex [23].

Definition 2 (Connectivity of Points): Let $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ be two data points in the range image. Two points are defined as connected if their corresponding voxel elements are 6-adjacent.

Definition 3 (Connected Point Sets): A point set $P = \{p_i\}$, $i = 1, \dots, m$ is connected if $\forall i, j$ ($i \neq j$), p_i and p_j are defined as connected.

Definition 4 (Connectivity of Connected Point Sets): Two connected point sets $P = \{p_i\}$, $i = 1, \dots, m$, and $Q = \{q_j\}$, $j = 1, \dots, n$, are defined as connected if $\exists p_i \in P$ and $q_j \in Q$ such that p_i and q_j are connected.

C. Reeb Graph Extraction Algorithm

The aim of the algorithm is to first describe the topology of a scene corresponding to a vehicle occupant using the Reeb graph extraction. For this purpose, one needs to first identify and extract the critical points of the Reeb graph where the change in topology occurs. Branches connecting the critical points represent different parts of the scene. The goal is that the resulting Reeb graph will have few branches, of which one corresponds to the upper torso of the person including the neck and head. Thus, the endpoints of the Reeb graph are used to identify potential head candidates.

The algorithm works as follows. As a Morse function, we choose the radial distance function that was proposed in [17]. This function is advantageous over other functions since it is rotationally invariant. It was also shown in [17] that this way, a Reeb graph can be obtained that is also invariant under scale and translation.

The flow of the Reeb graph extraction algorithm is shown in Fig. 5. It starts with the identification of an origin o of the

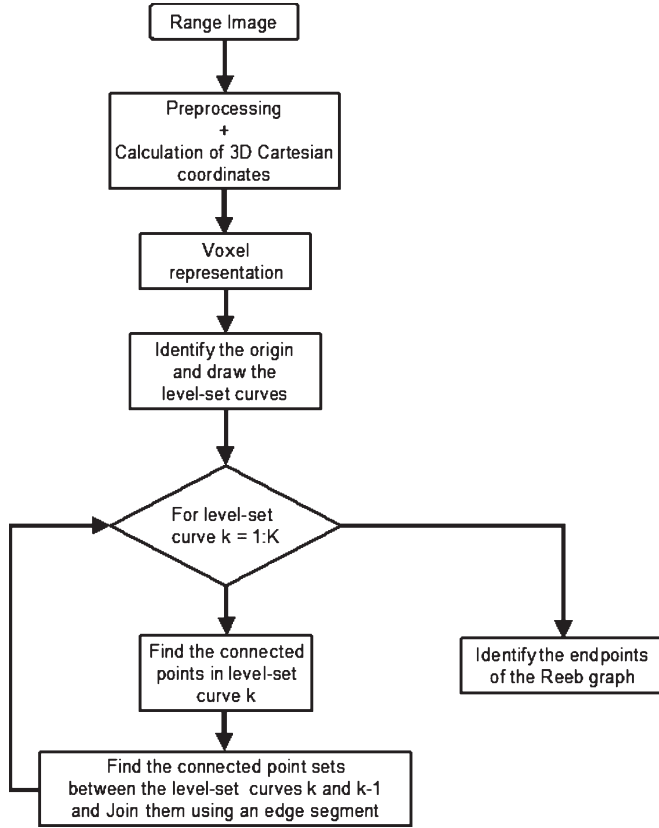


Fig. 5. Flow diagram of the Reeb-graph-based head-detection algorithm.

scene from where the distance function gradually grows in K steps $r_k = k \cdot \Delta r$, $k = 1, \dots, K$, of fixed step size Δr until all data are covered. The step size plays the role of resolution of the Reeb graph and has been chosen to be equal to the voxel size of 100 mm, which corresponds to the typical radius of a human's head.

In [17], the centroid of the object is considered as a starting point. Since our goal is to obtain a graph invariant under rotation of the upper torso of the person, we have chosen the point close to the hip of the passenger [see Fig. 4(b)] as the origin. This point is localized as follows. We first determine the range $[x_{\min}, x_{\max}]$ of values in which voxels are activated and choose the central x -coordinate ($x_c = (x_{\min} + x_{\max})/2$). Among all voxels with this x -value, the one with the minimum z -value is chosen as a starting point.

The level sets of the distance function are concentric spheres of radius r_k (see Fig. 6). In each of the K steps, we select the data points lying between spheres k and $k+1$, that is, the points having a distance $r \in [r_k, r_{k+1}[$ to the origin o . The voxel connectivity between these points is then determined as described in the previous section (see Definition 2), and then accordingly, the corresponding connected point sets are identified. Next, each connected point set is assigned with a cluster node at its centroid. At the end, the connectivity between the connected point sets at level i to those at level $i-1$ is established (see Definition 4), and the corresponding cluster nodes are joined by an edge segment. Along this procedure, a cluster node will be identified as a saddle critical point if it has at least two branches attached to it. Similarly, a cluster

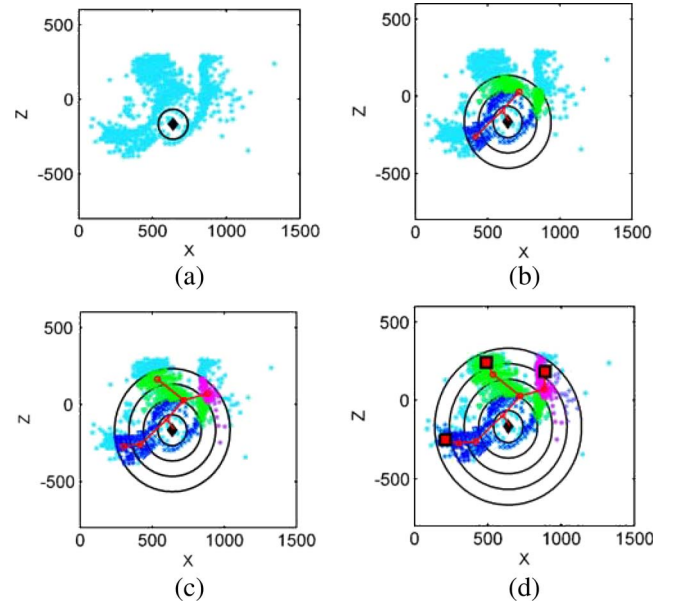


Fig. 6. Reeb graph extraction. (a)–(d) Level sets of radial distance function (plotted as circles) and the corresponding identified connected points in each level set. The connected point sets are joined by an edge segment. The shown images are the projection of the 3-D coordinates on the $x-z$ vehicle coordinate plane. (d) shows also the identified endpoints of the projected Reeb graph.

node will be identified as one of the extremal critical point (minimum or maximum) if it is the endpoint of the branch. Finally, the endpoints of the graph can be identified as potential head candidates. The position of the head candidate is then given by the coordinates of the corresponding cluster nodes.

IV. MULTIPLE HYPOTHESIS TRACKER

The proposed Reeb graph extraction gives a list of possible head candidates for each image frame. Once the possible head candidates (i.e., the endpoints of the Reeb graph) are identified and located, the MHT algorithm is applied on these candidates. The purpose of the MHT algorithm is then to automatically generate, out of the positions of all head candidates, the trajectory of the occupant's head.

The task of the MHT algorithm consists of three parts: 1) A data association part associates the head candidates in the instantaneous image frame to existing tracks; 2) a track filtering task updates a track by combining the predicted position of the tracked candidate with the associated instantaneous measurement and predicts the position in the next time step; and 3) finally, based on the topology information of the scene, the algorithm should automatically find, out of all tracked head candidates, the most likely head candidate, which is an estimate of the true head position.

The problem of handling multiple head candidates is, in general, difficult due to the following reasons: 1) Measurements given by the head detector contain noise; 2) the targets may not be detected in successive frames; and 3) some targets may move out of the image frame. The MHT algorithm is particularly useful in these cases as it is capable of dealing with missing detections, spurious measurements, and track initialization for new targets (see [19] and [24] for more details). In the following

sections, we briefly describe the data association approach and the tracking algorithm considered in this paper.

A. Data Association

In the MHT algorithm, each hypothesis represents a track, and it predicts the location of the head candidate in the next image frame. These predictions are compared with the actual head candidates detected in the next image frame on the basis of their Mahalanobis distance, and accordingly, a distance ambiguity matrix is calculated. The candidates are associated to the existing tracks based on this distance ambiguity matrix. Refer to [24] for more details.

The track initialization and termination issues of the MHT algorithm are solved as follows.

- 1) Track initialization: If a candidate could not be assigned to an existing track for more than one frame, a new track is started.
- 2) Track termination: A track is terminated if no candidate is associated with it for several frames. This number of frames has been chosen to be seven, which corresponds to half a second in time. The termination of the tracks guarantees that the maximum number of tracks stays limited over time.

B. Linear Kalman Filter

The tracker uses a Kalman filter with a nearly constant velocity model [25] to provide the filtering and the prediction of the tracks. Let $\mathbf{x} = [x, y, z]$ be the 3-D location of a head candidate and $\mathbf{V} = [V_x, V_y, V_z]$ be the corresponding velocity. The constant velocity model maps the (corrected) state $\mathbf{s}(k) = [\mathbf{x}(k), \mathbf{V}(k)]^T$ of a track at the k th time step to the (predicted) state at the next time step by the equation

$$\mathbf{s}(k+1) = \mathbf{A}\mathbf{s}(k) + \mathbf{v}(k) \quad (3)$$

$$\mathbf{x}(k) = [\mathbf{I} \ 0]\mathbf{s}(k) + \mathbf{w}(k) \quad (4)$$

with the state transition matrix $\mathbf{A}$ given by

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & T_{\text{cam}} & 0 & 0 \\ 0 & 1 & 0 & 0 & T_{\text{cam}} & 0 \\ 0 & 0 & 1 & 0 & 0 & T_{\text{cam}} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

where T_{cam} is the acquisition time of the camera. In this nearly constant velocity model, the acceleration of the head is treated as process noise $\mathbf{v}$, where the variance of the process noise has been chosen to cover the maximum acceleration to be expected in a car (prior to a crash), which is assumed to be 1 g. This choice for the variance of process noise will ensure that person movements like the one occurring due to precrash braking will be covered by the model. Our motivation for choosing this simple model was that accelerations like precrash braking suddenly occur and are therefore difficult to incorporate into a deterministic movement model. Other tracking models (for example, the interacting multimodels proposed in [4], which capture all the motions that may be experienced) would

probably improve the performance. However, we show that the considered model is sufficient to get adequate performance.

The measurement noise $\mathbf{w}$ is determined by the noise in the range measurement and has accordingly been chosen. The Kalman filter corrects the predicted position of a head candidate by updating it with the current measured position, thus filtering the trajectory of the head candidate.

C. Head Candidate Selection

The multiple-hypothesis tracker provides the trajectory of several head candidates. The last step in the head detection algorithm is to select, out of all the head candidates, the most likely estimate corresponding to the true head of the person. This task is accomplished by initially selecting a candidate according to a rule derived in Section V-D from the analysis of the topology of the scene by considering the geometrical constraints of the situation. This candidate is subsequently tracked. In case the track is terminated, a new candidate is selected. For details about the selection rule, see Section V-D.

V. EXPERIMENTS AND RESULTS

The evaluation methodology includes the performance of the two subcomponents of the head detection system: the head candidate detection using the Reeb graph technique and the MHT. Several experiments have been conducted to test the validity of the proposed approach. The following situations have been considered for the evaluation: 1) A passenger sits in a normal position; 2) a passenger bends forward i.e., the sequence consists of an occupant moving from in position to out of position (see Fig. 1); and 3) a passenger carrying headlike objects, for example, a balloon. A database of several sequences was recorded with two different passengers of different sizes, and they were asked to perform several tasks to take into account the movements described above. The database consists of 11 sequences with a total of 3300 images. To acquire the data, the camera was installed in the overhead module of a car (see Fig. 2) equipped with a computer and data-acquisition software. The capturing and the processing of images are simultaneously done, whereas the evaluation of performance of the algorithm is done offline.

The evaluation of the head detection system is organized as follows. First, the Reeb graph extraction is applied on all low-resolution range images, and we discuss what kind of different topologies can occur and to which occupancy situations they are related. In the second step, we evaluate if the head is always successfully detected by the Reeb graph. Therefore, the positions of the detected head candidates are compared with ground-truth measurements. Third, the MHT with a nearest neighbor data association and a linear Kalman tracker is applied on the possible head locations detected by the Reeb graph, and the resulting position is compared with the ground-truth data.

A. Topology of Extracted Reeb Graphs

The following possible situations of the scene that introduce variations in the Reeb graph were identified: the passenger head lies close to the headrest, the passenger bends forward, and the passenger carries any headlike objects. As examples, Fig. 7

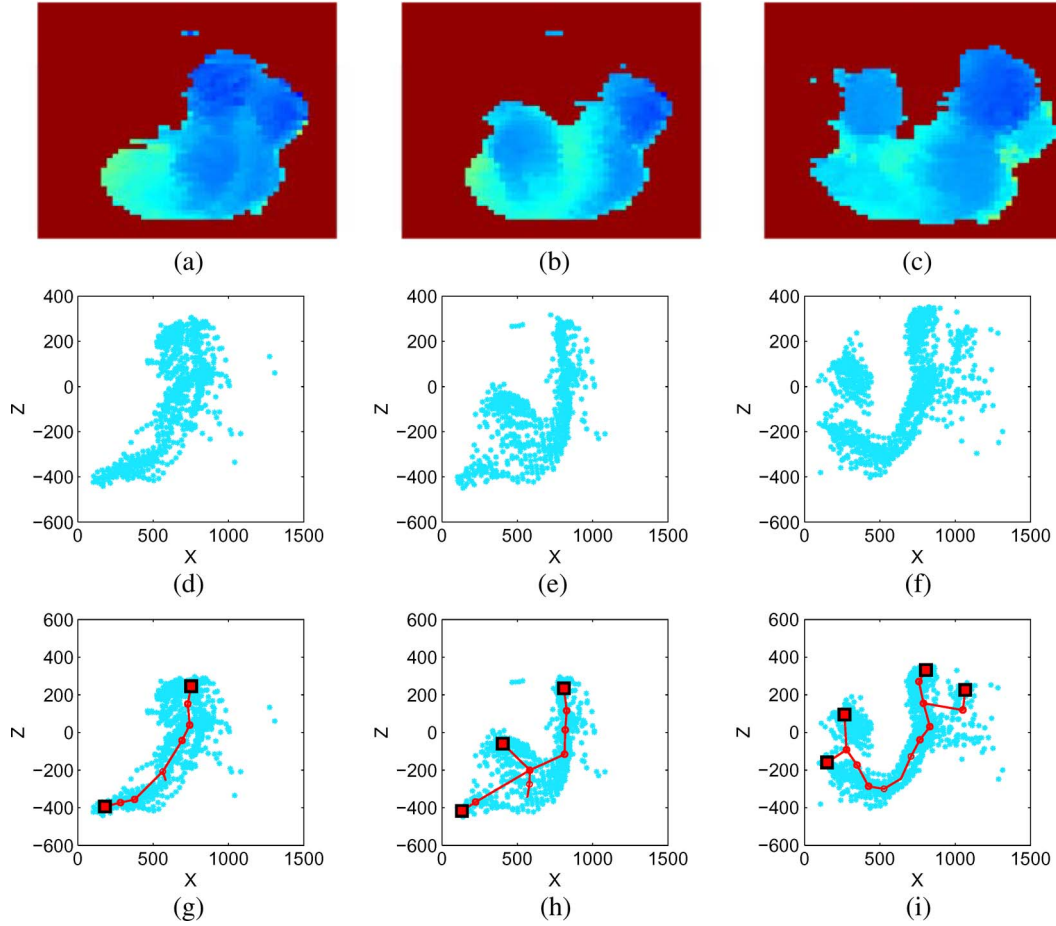


Fig. 7. (a)–(c) Snapshots of range images for different situations of the scene. (a) Passenger head lies close to the headrest; (b) passenger bends forward; and (c) passenger holding a balloon. The range images are shown in false-color representation. (d)–(f) Scatter plots of the coordinate points of the same situations shown in (a)–(c). The shown images are the projection of the 3-D coordinates on the $x - z$ vehicle coordinate plane. (g)–(i) Projected Reeb graphs at a resolution level, $\Delta r = 100$ mm.

shows three examples of range images of different situations of the scene. Fig. 7(a) depicts the situation where the passenger head lies close to the headrest, Fig. 7(b) shows the situation of a passenger bending forward, and Fig. 7(c) shows the situation where a passenger carries a balloon. Note that the shown images are range images where each pixel value represents the radial distance from the camera center to a point in the scene where the corresponding pixel is focused. The (x, z) -scatter plots of the corresponding coordinate points are shown in Fig. 7(d)–(f).

The Reeb graph of the situation in Fig. 7(a) is shown in Fig. 7(g), and the corresponding endpoints are marked with a square symbol. In this particular situation, the Reeb graph produces only two endpoints. Since the passenger head lies close to the headrest, they are considered as topologically connected and thus represented with only one branch. The other branch corresponds to the legs of the passenger. Fig. 7(h) shows the Reeb graph of the situation where the passenger bends forward as in Fig. 7(b). As expected, the Reeb graph produces three endpoints: one corresponds to the headrest, one corresponds to the head of the passenger, and the third one corresponds to the legs of the passenger. In the situation where the passenger bends more forward and is looking for something under the seat, the Reeb graph produces only one branch corresponding to the legs and head of the passenger.

For the case when the passenger is holding a ball, as in Fig. 7(c), an additional branch can be generated, as shown in Fig. 7(i). The Reeb graph has, in this case, four endpoints, representing the headrest, the passenger's head, the ball, and the lower torso of the passenger, respectively. In the situation where the balloon is on the lap, the Reeb graph has only one branch corresponding to the combined region of the balloon and the legs of the passenger since they are considered as topologically connected. Clearly, in all three situations, the Reeb graph always produces a branch leading to the passenger's head. Furthermore, the number of Reeb graph branches (and, subsequently, the number of head candidates) varies between two and four, as can be seen in Fig. 8, showing a histogram of the number of head candidates. The effect of varying the step size of the Reeb graph (see Section III-C) was already investigated in a previous paper [26], and it was observed therein that the chosen value of the step size is sufficient to capture the topology changes to successfully detect the head location.

B. Ground-Truth Information

To show that the Reeb graph can be used to localize the head position, the results of the head-detection algorithm are compared against the ground-truth data, which are acquired

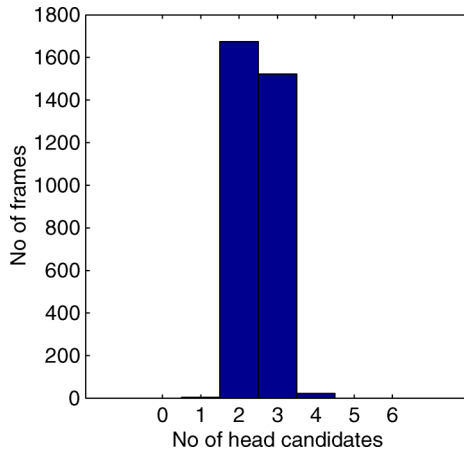


Fig. 8. Histogram of the number of Reeb graph endpoints (subsequently, the number of head candidates) over all images.

by 3-D string sensors. The 3-D string sensor consists of three variable-length cables that are attached to the potentiometers placed on distinct locations on the dashboard and in the ceiling of a car. The free extremities of the cables are linked on to each other with a belt. During the recordings, the occupants were asked to wear this belt on their forehead. The 3-D string sensor directly gives a measure of coordinates of the crossing point, i.e., the (x, y, z) -coordinate values of the crossing point at the forehead where three sensor cables were linked. This measure is then taken as the ground truth to compare with the coordinates of possible head locations given by the Reeb graph approach. Note that the string sensors do not have any influence on the range image as the cables are very thin and the potentiometers are not in the field of view of the camera. The calibration of the string sensor system showed that the accuracy in measuring the x -coordinate is (depending on the position) about 1 cm.

C. Evaluation of the Head-Detection Method

The head-detection algorithm is evaluated on all sequences, and results are then compared with the ground-truth measurements. A total of ten sequences consisting of 3225 frames of images were acquired for the evaluation. The sequences where the passenger was holding headlike objects were not included in the evaluation as in these cases, ground-truth information could not be obtained; objects would obstruct the string sensors. To verify if there is always a head candidate close to the ground-truth position, the notion of *successful detection* is considered as follows: if the Reeb graph of a particular scene produces n endpoints, i.e., $E_1, E_2, \dots, E_n$, respectively, and E_k ($1 \leq k \leq n$) is the closest endpoint to the ground-truth measurement, then the point E_k is said to be successfully detected only if it is within 150 mm from the ground-truth measurement. The value of 150 mm has been chosen as it corresponds to the average head radius of a person plus the maximum measurement noise. Since the ground truth gives the measure of a forehead point, it has been compared with the most forward point of the connected point sets instead of its centroid. This most forward point is calculated by computing the most forward point of an ellipsoid fitted to the connected point sets associated with the endpoints of the Reeb graph.

The successful detection results of the head detection algorithm are listed in Table I (column 3), and the corresponding histogram of difference error between the ground truth and the closest endpoint of the Reeb graph calculated over all sequences is shown in Fig. 9. Results show that the Reeb graph gives a branch leading to the occupant's head in most of the frames. This shows that the topology information of the scene is indeed very successful in detecting the position of the occupant's head while the number of possible head candidates is small.

There are, however, situations where the Reeb graph fails to give an endpoint closer than 150 mm to the occupant's head. Out of all situations, it is observed that 70% of the errors are due to the situation when the head is connected to the headrest leading to a common branch of the Reeb graph and the corresponding endpoint of this branch lies either on the head or on the headrest. In the latter case, the distances of the head to the endpoint of the graph may be larger than 150 mm.

However, this situation occurred not for the entire sequence but only for few frames. Moreover, this situation is not critical since the head is far from the airbag module in these cases and thus need not precisely be located.

Furthermore, from the geometrical constraints of the situation in a car, the following rule has been derived that allows a further reduction in the number of candidates.

Rule 1 (Reduction of the Head Candidates): Assuming that the only endpoint that can be higher (in z) than the head can be the headrest implies that it is sufficient to consider only two candidates with the largest z -position.

D. Evaluation of the Multiple-Hypothesis Head-Tracking Algorithm

The aim of the MHT algorithm is to generate the trajectories of head candidates from the instantaneous head candidates and to select the most likely estimate of the true candidate. The MHT algorithm was tested on all sequences considered in the previous section. Note that in these sequences, the passenger was mainly asked to move from in position to out of position and return back to in position. For each sequence, possible head locations were automatically extracted for each image frame using the head detection algorithm based on the Reeb graph technique. This method provides the most forward point of the ellipsoid fitted to the connected point sets that correspond to the region of endpoints of the Reeb graph. The MHT is applied on these measurements to handle the ambiguities in associating the measurements with predictions. The Kalman filter states of the valid tracks are the final output of the head-tracking algorithm.

To select the correct head candidate, the following rule has been applied on the valid hypothesis.

Rule 2 (Correct Candidate Selection): In case of only two candidates, as in Fig. 7(g), the algorithm chooses the candidate that has the largest z -value as the correct head candidate, discarding the candidate corresponding to the legs of the person. If there are three or more candidates, as in Fig. 7(h), the algorithm first selects the two candidates with the largest z -value (based on rule 1), and among these two, it chooses the one that is most forward in the x -direction.

TABLE I
HEAD DETECTION AND MHT RESULTS

Total no of sequences (Total no of frames)	Average number of Reeb graph end-points for each frame	Number of frames successfully detected (%head detection results)	Successful detection if only two candidates are considered	Number of frames correctly detected (%MHT results)
10(3225)	2.5	3114(96.56%)	3113(96.53%)	3224(99.96%)

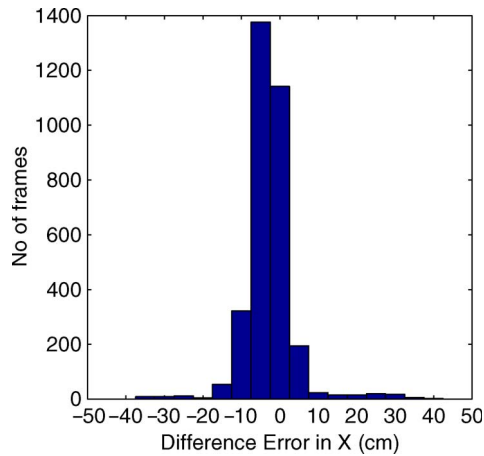


Fig. 9. Head detection results. Histogram of the difference between the ground-truth measurement and its closest endpoint of the Reeb graph over all frames.

This way, the candidates corresponding to the legs of the passenger are completely discarded. Note that it would not be sufficient to always simply select the candidate with the largest z -value, as for small persons and children, the headrest may result in a candidate with a larger z -value. Rule 2 also handles this situation and selects only the candidate corresponding to the occupant's head as the head is always located more forward in the x -direction than the head rest.

Rule 2 will be applied for the first frame to initialize the correct head candidate; subsequently, the track of the selected candidate will be considered as the final head candidate. In case the selected track is terminated, rule 2 will be applied to reinitialize the selection of the correct head candidate. In this process, rule 1 will be applied only if there are more than two candidates; rule 1 selects in that case the two candidates highest in z -position, and rule 2 will select the one that is most forward in the x -position. However, note that these rules will be used only for the first frame to initialize the correct head candidate location.

The MHT algorithm has the ability to create new tracks for objects entering the image frame and terminate tracks for objects exiting the image frame. It can also handle occlusions, that is, continuation of a track for few defined numbers of frames even if it does not find any measurements associated to it. We show one such situation where the MHT algorithm successfully handles the creation and termination of tracks when there is a topology change occurring in the scene. In Fig. 10, a few frames from one of the test sequences are shown, together with the tracking results. The ellipse symbols denote the identified tracks. Note that the identified tracks are the output of the MHT algorithm before applying rules 1 and 2. First, the application of rule 1 reduces the number of head

candidates to be considered for the tracking, and then, rule 2 will further ensure that the Reeb graph endpoint corresponding to the head will be selected.

Fig. 11 shows the x -coordinate of the resulting trajectories (plotted in solid lines) when the MHT data association filter is applied on the entire sequence shown in Fig. 10. Fig. 11 also shows the raw output of the head candidates (plotted in points) before filtering is applied. Note that the tracking of each hypothesis is done in the 3-D coordinate space, although it shows only the x -coordinate that is relevant for the out-of-position detection. The algorithm initially starts with two tracks when the occupant leans on the backrest with the head close to the headrest [as shown in Fig. 10(a)]. The algorithm then splits the existing track corresponding to the region of the headrest and the occupant's head into two different tracks as soon as the occupant's head and the headrest of the seat are clearly separable. Of these two new tracks, one corresponds to the occupant's head, and the other corresponds to the headrest. From that moment on, the candidate corresponding to the legs could be discarded according to rule 1, which is derived in Section V-C.

In a special case when the passenger bends extremely forward, the Reeb graph assigns only one branch to the region corresponding to the upper torso and legs of the passenger. In this situation, the track corresponding to the legs of the occupant was terminated by the MHT algorithm [as the legs and upper torso of the passenger are topologically connected in this case; see Fig. 10(c)], and a new track was started as soon as an additional branch becomes available [see Fig. 10(d)]. It is apparent from these figures that there is always a tracker that follows the true occupant's head position.

To evaluate the performance of the MHT algorithm, the results are compared with the ground-truth measurement. The final head candidate estimated by the MHT algorithm after rule 2 has been applied on all the identified tracks is said to be *correct* if it is within 150 mm from the ground-truth measurement. The x -values of the final head candidate (shown in solid line) and ground-truth data (shown in dotted line) are plotted in Fig. 12. It can be observed that the x -values are decreasing as the occupant is moving close to the airbag control unit. Clearly, the final head candidate closely follows the ground-truth data. The correct detection results of the MHT algorithm after rule 2 has been applied are listed in Table I (column 5), and the corresponding histogram of difference error between the final head candidate and the ground truth is shown in Fig. 13. With an overall detection rate of 99.96%, the MHT tracking algorithm is very successful. Clearly, the MHT algorithm not only automatically follows the correct head location but also alleviates the detection errors that occur due to noise and occlusions.

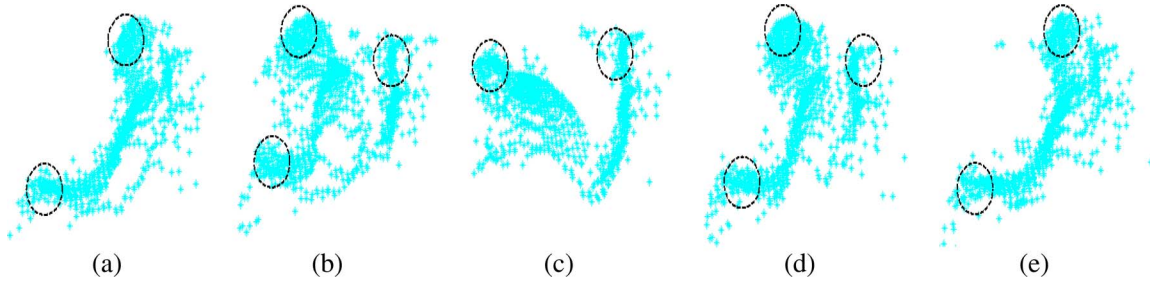


Fig. 10. (a) First, (b) 50th, (c) 120th, (d) 192th, and (e) 221st frames of the sequence considered for the evaluation. The MHT results are denoted by ellipses. Note that the shown ellipses are the identified tracks before applying rule 2.

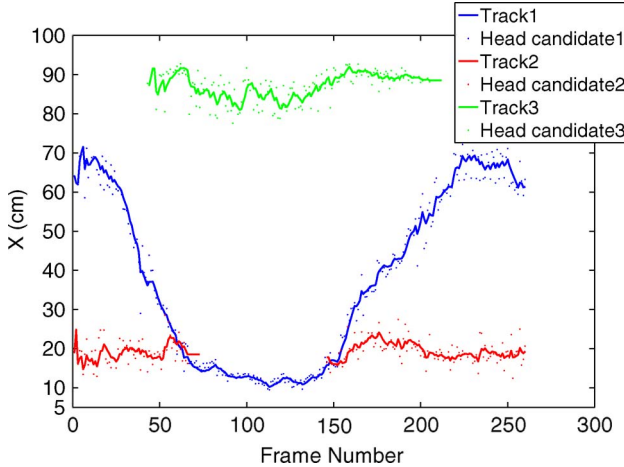


Fig. 11. Trajectories of the MHT data association filter applied on one sequence. The points are the raw output of the head candidate positions, and the lines are the corresponding filtered positions.

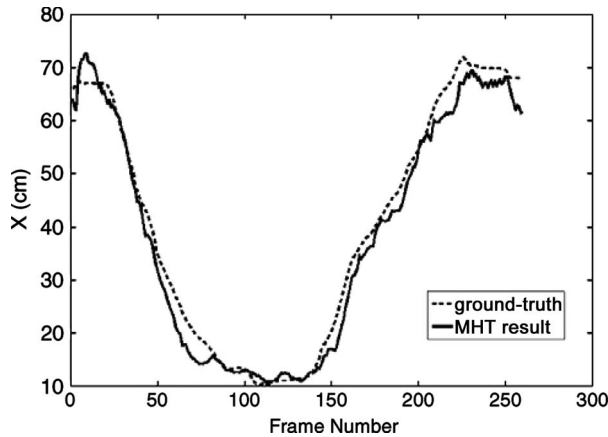


Fig. 12. Tracking results. The final correct head candidate estimated by the MHT algorithm (shown in a solid line) is compared with the ground-truth measurement (shown as a dotted line). The points shown in red correspond to the raw measurements before filtering is applied.

VI. PROCESSING REQUIREMENTS

A. General Remarks

In this section, we present an estimate of the processing requirements of the proposed topology-based head detection and tracking algorithm. In particular, the processing time requirements for the addressed real-time application could be rather hard, as the frame rate of the system has to be 40 Hz [4]. In addition, the economical constraints of the automotive

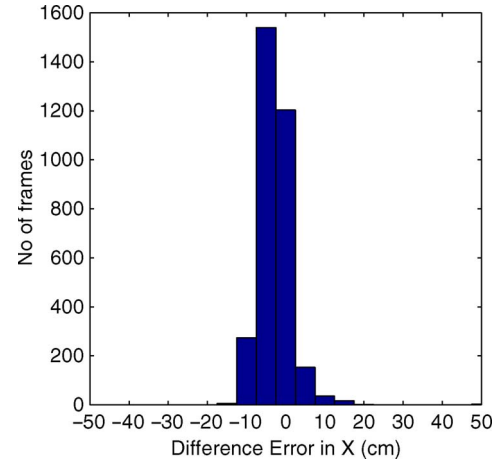


Fig. 13. Multiple-hypothesis tracker results. Histogram of the difference between the ground-truth measurement and the correct head candidate given by the MHT filter.

TABLE II
PARAMETER DETERMINING THE PROCESSING REQUIREMENTS
OF THE REEB GRAPH EXTRACTION

Parameter	Notation	Value
Number of pixels	N_{pix}	2600
Number of voxels	N_{vox}	$15 \times 8 \times 10 = 1200$
Number of levels	N_{level}	≤ 8
Number of centroids in each shell	N_{cen}	≤ 4

industry have to be considered in the choice of the processing unit.

Most of the processing power for the proposed algorithm is needed for the Reeb graph extraction. Therefore, we will, in the following, concentrate on evaluating this part. As the number of pixels of the proposed camera is only $N_{pix} = 52 \times 50 = 2600$, the preprocessing, consisting of a filtering and coordinate calculation, is relatively negligible. Furthermore, the track processing takes only few resources, where the most time-consuming operation is the inversion of matrices that are of dimension 6×6 as the state vector has the length 6 (compare Section IV-B). The other parameters that determine the processing requirements of the tracking are the number of measurements and the number of tracked states. However, it has been demonstrated in Section V-D that the maximum number of tracks can be limited to four without loss of performance. Therefore, the computational effort for the tracking is negligible compared with that of the Reeb graph extraction.

TABLE III
NUMBER OF OPERATIONS FOR THE INDIVIDUAL STEPS OF THE REEB GRAPH EXTRACTION

Algorithm Step	Order of Operations	Comments
1. Voxel activation 1.1. Assign pixel to voxel 1.2. Compute activation level 1.3. Compare voxel activation with threshold	$4N_{pix}$ $2N_{pix}$ N_{vox}	3D-histogramming Summing distance squares
2. Determination of origin	$6N_{vox}$	Minimum and maximum determination in x , then minimum in z
3. Assigning voxels to level sets	$10N_{vox}$	Calculating Euclidean distances, division by step size
4. Connectivity inside each level 4.1. Determination of connected points 4.2. Label points according to their connectivity 4.3. Calculate centroids	$6N_{vox}$ $N_{level} \left(\frac{N_{vox}}{N_{level}} \right)^2$ $3(N_{pix} + N_{cen}N_{level})$	Check for each voxel which of its 6 neighbours is activated and in the same level Assign identical labels to connected points in a level, relabel already labeled points, if needed Calculate for each shell and connected component the center of gravity
5. Connectivity between levels 5.1. Determine point connectivities between levels 5.1. Determine end points of Reeb graph	$3N_{vox}$ $N_{cen}N_{level}$	Check for each voxel which of the 6 neighbours is activated and in the adjacent level Check on each level which point set is connected to a lower level

B. Processing Requirements of the Reeb Graph Extraction

The parameters determining the processing requirement of the Reeb graph extraction are listed in Table II. The most important parameter in the Reeb graph extraction is the number of voxels N_{vox} , which is determined by the compartment volume on the passenger side (assumed to be 1500 mm $\times$ 800 mm $\times$ 1000 mm) and the resolution, which was chosen to be 100 mm in either direction.

Table III gives an estimate of the number of operations for each step of the Reeb graph extraction. One recognizes that the most time-consuming task is clearly the determination of connected point sets in each individual level set. The number of operations in this step is mainly determined by the number of voxels in each level set, which is on the order of N_{vox}/N_{level} , the ratio of the total number of voxels to the number of levels considered. The number of operations for this step is 0.18 MOP/frame if the parameters are chosen according to Table II, while the other operations account in sum for less than 0.02 MOP/frame. Assuming a frame rate of 40 Hz, the processing power would be less than 8 MOPs.

These rather small numbers mainly result from the low resolution on which the Reeb graph is extracted. Both the number of pixels and the number of voxels are on the order of 100 smaller than the number of pixels in a video graphics array image. Consequently, the computation requirements for the proposed Reeb graph extraction are two orders of magnitude lower than the performance modern embedded controllers can offer, which are specially designed for real-time video processing (see the discussion in [4]).

VII. CONCLUSION

This paper has presented a general topology-based framework for low-resolution range images that offers a systematic way to detect and track a vehicle occupant's head. The proposed

approach explores the topology information of a scene represented in a 3-D skeleton to detect the position of the occupant's head. The Reeb graph technique is considered for this purpose. However, any topology representation, other than the Reeb graph, could also be useful. Previous approaches to Reeb graph extraction are applied to a dense 3-D representation of a closed object surface, e.g., a triangulated mesh representation. Here, we have adapted the Reeb graph concept to low-resolution range images. This paper has been motivated by the work of Xiao *et al.* [18], who proposed a DRG approach to unorganized clouds of 3-D points. To cope with the particular challenges here, such as the noise and the large variations in the density of the data, we have proposed a voxel neighborhood connectivity notion for building the Reeb graph. Invariance of the graph under rotations has been achieved by using a Morse radial distance function.

The approach for head detection has been applied on several sequences, which were recorded taking into account the different possible variations of the scene. The performance of the head detection algorithm using the Reeb graph technique has been compared with the ground-truth data, which are acquired with string sensors. Results have shown that the Reeb-graph-based head detection can successfully locate the occupant's head in a wide range of applications. It has been shown that on average, only 2.5 head candidates result from each image, with the maximum being four candidates. In addition, we have shown that it is possible to derive from the topological situation and geometrical constraints of the situation a simple rule that reduces the number of candidates that must be considered in the tracking to a maximum of two.

To filter the noise in measurements from head candidates, to associate measurements to tracks, and to select the right head candidate out of all the head candidates given by the Reeb graph, an MHT algorithm has been applied. One of the main strengths of the proposed topological approach is that it yields a maximum of four candidates, of which only two need to

be considered for the tracking. Therefore, the common issue of combinatorial explosion in MHT will not be a problem. Results of the MHT algorithm have shown that close to 100% performance could be achieved in detecting the correct head candidate position.

The proposed method has been implemented in MATLAB, and the tests presented herein have been evaluated offline. Therefore, a next step in the development and evaluation of our method would be a real-time implementation of our proposed approach allowing online testing. Clearly, more comprehensive tests have to be performed, as well as a comparison with other approaches based on geometrical shapes, before concluding that our method is superior to others under all conditions. Furthermore, to comment on the technical feasibility of the presented approach, we have evaluated the computational requirements of our approach and shown that they are far below what can be achieved with (automotive qualified) embedded controllers that are specially design for real-time video processing.

ACKNOWLEDGMENT

The authors would like to thank Prof. H. Krim and Dr. S. Baloch of North Carolina State University, Raleigh, for the fruitful discussions on the Reeb graph approach.

REFERENCES

- [1] M. E. Farmer and A. K. Jain, "Occupant classification system for automobile airbag suppression," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2003, vol. 1, pp. I-756-I-761.
- [2] K. S. Huang and M. M. Trivedi, "Video arrays for real-time tracking of person, head, and face in an intelligent room," *Mach. Vis. Appl.*, vol. 14, no. 2, pp. 103-111, Jun. 2003.
- [3] M. M. Trivedi, S. Y. Cheng, E. M. C. Childers, and S. J. Krotosky, "Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1698-1712, Nov. 2004.
- [4] M. E. Farmer and A. K. Jain, "Smart automotive airbags: Occupant classification and tracking," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 60-80, Jan. 2007.
- [5] P. R. Devarakota, M. Castillo-Franco, R. Ginhoux, B. Mirbach, and B. Ottersten, "Occupant classification using range images," *IEEE Trans. Veh. Technol.*, vol. 56, no. 4, pp. 1983-1993, Jul. 2007.
- [6] H. Kong, Q. Sun, W. Bauson, S. Kiselewich, P. Ainslie, and R. Hammoud, "Disparity based image segmentation for occupant classification," in *Proc. IEEE Comput. Vis. Pattern Recog. Workshop Object Tracking Classification Beyond Visible Spectrum*, Jun. 2004, pp. 126-132.
- [7] A. Douha, F. Cutzu, R. Hammoud, and S. Kiselewich, "Triangulation based technique for efficient stereo computation in infrared images," in *Proc. IEEE Intell. Veh. Symp.*, Columbus, OH, Jun. 2003, pp. 673-678.
- [8] M. Fritzsche, C. Prestele, G. Becker, M. Castillo-Franco, and B. Mirbach, "Vehicle occupancy monitoring with optical range-sensors," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2004, pp. 90-94.
- [9] S. J. Krotosky, S. Y. Cheng, and M. M. Trivedi, "Real-time stereo-based head detection using size, shape and disparity constraints," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 550-556.
- [10] Z. Hu, T. Kawamura, and K. Uchimura, "Grayscale correlation based 3D model fitting for occupant head detection and tracking," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2007, pp. 1252-1257.
- [11] T. Tung and F. Schmitt, "The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes," *Int. J. Shape Model.*, vol. 11, no. 1, pp. 91-120, 2005.
- [12] S. Biasotti, B. Falcidieno, and M. Spagnuolo, "Extended Reeb graphs for surface understanding and classification," in *Proc. 9th Int. Conf. DGCI*, Uppsala, Sweden, Dec. 2000, vol. 1953, pp. 185-197.
- [13] Y. Shinagawa and T. L. Kunii, "Constructing a Reeb graph automatically from cross sections," *IEEE Comput. Graph. Appl.*, vol. 11, no. 6, pp. 44-51, Nov. 1991.
- [14] G. Reeb, "Sur les points singuliers d'une forme de pfaff completement integrable ou d'une fonction numerique [on the singular points of a completely integrable pfaff form or of a numerical function]," *Compte Rendus Acad. Sci. Paris*, vol. 222, pp. 847-849, 1946.
- [15] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Aug. 2001, pp. 203-212.
- [16] C. Menier, E. Boyer, and B. Raffin, "3D skeleton-based body pose recovery," in *Proc. 3rd Int. Symp. 3D Data Process., Vis., Transmiss.*, Jun. 2006, pp. 389-396.
- [17] S. Baloch, H. Krim, I. Kogan, and D. Zenkov, "Rotation invariant topology coding of 2D and 3D objects using Morse theory," in *Proc. IEEE ICIP*, Sep. 2005, vol. 3, pp. 796-799.
- [18] Y. Xiao, P. Siebert, and N. Werghi, "A discrete Reeb graph approach for the segmentation of human body scans," in *Proc. 4th Int. Conf. 3DIM*, 2003, pp. 378-385.
- [19] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Int. J. Comput. Vis.*, vol. 10, no. 1, pp. 389-396, Feb. 1993.
- [20] R. Lange, "Demodulation pixels in CCD and CMOS technologies for time-of-flight ranging," in *Proc. SPIE Conf. Sens., Cameras Syst. Sci. Ind. Appl.*, San Jose, CA, 2001, pp. 177-188.
- [21] J. W. Weingarten, G. Gruener, and R. Siegwart, "A state-of-the-art 3D sensor for robot navigation," in *Proc. IEEE IROS*, Sep. 2004, pp. 2155-2160.
- [22] R. Lange, "3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Siegen, Siegen, Germany, 2000.
- [23] S. N. Srihari, "Representation of three-dimensional digital images," *ACM Comput. Surv.*, vol. 13, no. 4, pp. 399-424, Dec. 1981.
- [24] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, Jan. 1988.
- [25] M. G. T. Huang and K. S. Trivedi, "Driver's view and vehicle surround estimation using omnidirectional video stream," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2003, pp. 444-449.
- [26] P. R. Devarakota, M. Castillo-Franco, R. Ginhoux, B. Mirbach, and B. Ottersten, "Application of the Reeb graph technique to vehicle occupant's head detection in low-resolution range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshop*, Minneapolis, MN, 2007, pp. 1-8.



Pandu Rangarao Devarakota (S'07) received the M.Tech degree in instrument technology from Indian Institute of Technology, New Delhi, India, in 2003 and the Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden, in 2008. As a part of his master studies, he was an exchange student with the Institute of Communication Technology, University of Karlsruhe, Karlsruhe, Germany. His Ph.D. program was in cooperation with the University of Luxembourg and IEE S.A., Contern, Luxembourg.

He is currently a Member Technical Staff with the Computer Aided Diagnosis Research Group, Siemens Information Systems Ltd., Bangalore, India.



Marta Castillo-Franco was born in Barcelona, Spain, in 1978. She received the M.S. degree in telecommunication engineering from the Universitat Politècnica de Catalunya, Barcelona, in 2001 and the Master's degree in image processing and artificial intelligence from the Ecole National Supérieure des Télécommunications de Bretagne, Brest, France, in 2002.

She started working in the automotive industry on the research and development of intelligent optical systems for safety applications. She is currently a Program Manager with the Advanced Engineering Department, IEE S. A., Contern, Luxembourg.



Romuald Ginhoux (S'00–M'04) was born in Dijon, France, in 1977. He received the M.Sc. degree from the Ecole Nationale Supérieure de Physique, Strasbourg, France, in 2000 and the Ph.D. degree in robotics from the University of Strasbourg in 2003.

Since 2006, he has been an R&D Engineer with MEDIAN Technologies, Sophia-Antipolis, France, working on computer-aided diagnosis software for radiology. His fields of interest include robotics and computer vision for medical devices and automotive safety systems.

Dr. Ginhoux is the recipient of the 2005 King-Sun Fu Memorial Best IEEE TRANSACTIONS ON ROBOTICS Paper Award.



Bruno Mirbach received the Ph.D. degree in theoretical physics from the University of Kaiserslautern, Kaiserslautern, Germany, in 1996.

He was Postdoctoral Researcher with the Center for Nonlinear and Complex Systems, Como, Italy, the Max-Planck-Institute of the Physics of Complex Systems, Dresden, Germany, and the University of Ulm, Ulm, Germany. Then, he joined the automotive industry, working on the research and development of intelligent optical systems for safety applications. He is currently the Senior Computer Vision Engineer

with IEE S. A., Contern, Luxembourg.



Serge Kater was born in Luxembourg in 1970. He received the Bachelor's degree in electrical engineering from the Institut supérieur de Technologie, Luxembourg, in 1997 and the Postgraduate Diploma in software development from the Open University, Milton Keynes, U.K., in 2008, where he is currently working toward the Master's degree in software development.

From 1997 to 2001, he was with Husky Injection Molding Systems as an Electrical Designer. From 2001 to 2003, he was with Waagner-Biro

Luxembourg as a Software Engineer. He is currently a Software Engineer with IEE S.A., Contern, Luxembourg.



Björn Ottersten (S'87–M'89–SM'99–F'04) was born in Stockholm, Sweden, in 1961. He received the M.S. degree in electrical engineering and applied physics from Linköping University, Linköping, Sweden, in 1986 and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1989.

He has held research positions with the Department of Electrical Engineering, Linköping University, the Information Systems Laboratory, Stanford University, and the Katholieke Universiteit Leuven,

Leuven, Belgium. From 1996 to 1997, he was the Director of Research with ArrayComm Inc., San Jose, CA: a start-up company based on his patented technology. He was with the Luxembourg International Advance Studies in Information Technology, Luxembourg. Since 1991, he has been a Professor of signal processing with the Royal Institute of Technology, Stockholm, Sweden, where he was the Head of the Department for Signals, Sensors, and Systems from 1992 to 2004, was the Dean of the School of Electrical Engineering from 2004 to 2008, and is currently with the Signal Processing Group, School of Electrical Engineering. He is the Director of securityandtrust.lu, University of Luxembourg. He is currently the Editor-in-Chief of the *EURASIP Signal Processing Journal* and a member of the editorial board of the *EURASIP Journal of Applied Signal Processing*. His research interests include wireless communications, stochastic signal processing, sensor array processing, and time series analysis.

Dr. Ottersten is a Fellow of the European Association for Signal Processing. He received an IEEE Signal Processing Society Best Paper Award in 1993, 2001, and 2006. He is a first recipient of the European Research Council advanced research grant. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and on the editorial board of the *IEEE Signal Processing Magazine*.