Using Multi-Modal 3D Contours and Their Relations for Vision and Robotics

Emre Başeski[*,a], Nicolas Pugeault[b], Sinan Kalkan[c], Leon Bodenhagen[a], Justus H. Piater[d], Norbert Krüger[a]

[a]*The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark*
[b]*Center for Vision, Speech and Signal Processing, University of Surrey, Guildford, United Kingdom*
[c]*Department of Computer Engineering, Middle East Technical University, Ankara, Turkey*
[d]*Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium*

Abstract

In this work, we make use of 3D contours and relations between them (namely, coplanarity, cocolority, distance and angle) for four different applications in the area of computer vision and vision-based robotics. Our multi-modal contour representation covers both geometric and appearance information. We show the potential of reasoning with global entities in the context of visual scene analysis for driver assistance, depth prediction, robotic grasping and grasp learning. We argue that such 3D global reasoning processes complement widely-used 2D local approaches such as bag-of-features since 3D relations are invariant under camera transformations and 3D information can be directly linked to actions. We therefore stress the necessity of including both global and local features with different spatial dimensions within a representation. We also discuss the importance of an efficient use of the uncertainty associated with the features, relations, and their applicability in a given context.

Key words: Cognitive vision, 3D contours, contour relations

1. Introduction

Global entities such as visual contours and their relations have a substantial importance in computer vision and robotics (see, e.g., [1, 2, 3, 4]) since they `a)` provide a semi-global overview of a scene, `b)` give more information than local features about the shape of an object [5] and `c)` are flexible enough for tasks such as classification and recognition [1, 2, 6]. Their potential for different vision-based tasks has been studied especially in 2D. In this work, we make use of a 3D contour representation that is multi-modal in the sense that it covers geometric as well as appearance information. We argue that such global reasoning processes complement widely-used local approaches and provide reliable geometric information for tasks that require 3D information (e.g., grasping).

Depending on the perceptual context, local and global aspects of visual entities play complementary roles. 2D local features such as SIFT [7] are known to be very robust in certain contexts such as object identification. However, they heavily rely on texture and do not give information about the shape of the object (see, e.g., [7, 8, 9]). These approaches also face two fundamental problems when they want to make use of relations (such as coplanarity or distance) between features (see, e.g., [10]). First, since a large number of local features is usually required to code an object or a scene, the number of second-order relations between these features grows exponentially and so does the resource requirements in terms of memory and computing time. Second, they do not provide explicit and interpretable information which is required for semantic interpretation of a scene. Also, the reliability of local 2D features is limited when objects need to be manipulated since they are difficult to relate with geometrical properties such as 3D shape. This limitation is mitigated when using 3D features (see, e.g., [11, 12, 13]) since 3D information can be directly related to actions.

As discussed in, e.g., [5, 6], global entities complement the local approaches by providing a more global overview of the scene. By combining local entities (in our case, local edge descriptors) into global entities (in our case, contours) the number of relations is reduced and hence the reasoning about the geometry and shape becomes easier. Also, global reasoning processes that take place in 3D are independent from view-point transformations, allowing for a significant reduction of complexity in any further analysis.

The importance of contours in human vision has been studied extensively (see, e.g., [14, 15, 16, 17]), revealing the fact that perceptual organization is a very important cue in human vision. What makes contours important for vision is not only the way the local features are grouped together but also the spatial relations between these contours. For example, certain relations

[*]Corresponding author
Email addresses: `emre@mmmi.sdu.dk` (Emre Başeski), `n.pugeault@surrey.ac.uk` (Nicolas Pugeault), `skalkan@ceng.metu.edu.tr` (Sinan Kalkan), `lebo@mmmi.sdu.dk` (Leon Bodenhagen), `Justus.Piater@ULg.ac.be` (Justus H. Piater), `norbert@mmmi.sdu.dk` (Norbert Krüger)

between groups of features (such as parallelism) are found to be more salient [3] than others and shown to be non-accidental (see, e.g., [5]). Also, in [18, 19] it has been shown that scrambled objects are difficult to recognize since the spatial organization of the contours is lost. Note that most of the studies on contours focus on the 2D aspects. Recently, Masayuki et al. showed in [20] the importance of 3D contours in human visual processing by demonstrating that, collinear line elements in 3D space are more salient than non-collinear elements.

2D visual contours and their relations have been used in computer vision and robotics in various contexts: In [21], contour relations are used as features for object recognition. Similarly, Henricsson [22] makes use of geometrical relations like proximity, curvilinearity and symmetry between contours to describe objects by combinations of these relations. These non-accidental contour relations have also been used by Dickinson et al. [23] to create aspect hierarchies, which are then used for recovery and recognition of 3D objects from a single 2D image.

For part-based representations, such as geons [24], the basic-level representation is composed of parts and their interrelations, and these parts are created by using non-accidental contour relations such as parallelism and symmetry. Similarly, Shapiro et al. [25] uses a relational model for describing 3D objects in terms of relations between simple parts such as sticks, plates and blobs. In more recent studies such as [1, 4], similar ideas have been applied to recognition and classification. Also, Fidler et al. [26, 27, 28] proposed a system to learn a contour-based hierarchy of parts for the representation and categorization of objects. A review of contour-based methods for classification can be found in [2]—note that all these approaches are in 2D.

In the context of robotics, contours have been used to infer grasping hypotheses for unknown objects. For example: [29, 30] approached the problem using 2D contours; more recently, [31] used 3D contours. The essential difference between our work and [31] is that we create 3D contours from local 3D features whereas 3D contours in [31] are inferred from 2D contours.

The advantages of using 3D contours and their relations can be summarized as follows:

View-Point Invariance: Many 3D relations (such as distance between entities) and properties (such as curvature) are view-point invariant: for example, if two 3D contours are parallel in one view, they will be parallel in any other view.

Reliability: For tasks such as grasping, the semi-global nature of contours makes 3D information more reliable than local 3D information in terms of geometry since global geometrical computations (e.g., slope or curvature calculation) are more robust to noise.

Saliency: Combinations of 3D features can create very rare and specific structures that are more salient than individual components. These constellations can reduce the search space when searching for relevant entities (e.g., finding cocolor and coplanar entities that can be grasped). An extended discussion on saliency can be found in [5].

Semantic Interpretation: Visual entities are expressed by explicit and interpretable parameters which themselves are embedded in a rich structural context. In this work, contours have well defined properties with clear geometrical and appearance interpretation. These parameters are then used to define relations between contours that are explicit and interpretable as well. This allows in particular for a direct link to actions.

Despite its advantages, the use of 3D contours and their relations faces some fundamental problems and these issues should be taken care of for an effective use of 3D contours.

Cardinality: When relations between contours are used as features, the number of features representing the object increases proportional to the number of combinations of contours that create the relations. For n visual features we need to calculate $n(n-1)/2$ relations. Note that cardinality is more problematic for local features and the usage of contours reduces the effect of this problem.

Early Commitment: Grouping local features into bigger groups with respect to certain criteria implies making some decision at an early stage of visual processing. Such an early commitment is brittle, due to ambiguity and noise in low level vision, and may adversely affect higher level reasoning.

Local Uncertainty: Reconstruction uncertainties of local features propagate to the contour they generate, and this uncertainty can degrade the performance of analytical estimates of 3D entities critically. For this reason, uncertainty needs to be estimated and to be taken in consideration in all processes.

Global Uncertainty: When dealing with 3D entities with strongly anisotropic uncertainty distributions (e.g., when extracted by stereo), there is a need to also code the limits until when 3D contours and their relations can be applied. For example, in an outdoor scene it does not make sense to talk about the distance of objects that are far away since stereo fails to give reliable information.

Note that, although 3D features suffer from higher uncertainty than 2D features, this comes with the benefit of high invariance to pose changes. As discussed in this article, this greater uncertainty needs to be properly modeled and handled to allow for reasoning in 3D. We argue that qualities of 2D and 3D representations are complementary, and both are required for the design of a robust vision system. Such a system should make decisions on weighting the use of 2D and 3D information according to the reliability of the 3D information which is modeled as the feature uncertainty.

The aim of this article is to exemplify the potential of 3D contours and second-order 3D contour relations on four different tasks and describe the particular requirements for the representation of the contours and their relations when using them for these tasks. We make use of four relations, namely, coplanarity, cocolority, distance and angle, which are important in the contexts of robotics and scene interpretation. Note that more relations can be defined for different contexts, however the aim of this article is not to give a complete set of relations but to show the importance of 3D contours and their relations in different domains. We discuss, in particular, how uncertainty of visual data can be handled in a geometrical reasoning processes.

The main aspects of our approach can be summarized as follows:

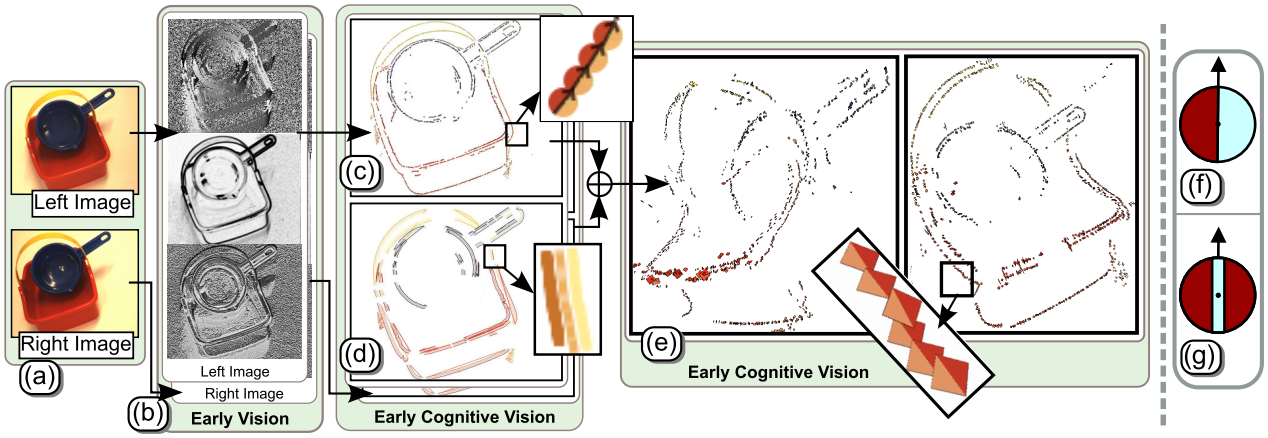

Figure 1: An overview of the hierarchical visual representation. **(a)** Stereo image pair, **(b)** Filter results, **(c)** 2D primitives, **(d)** 2D contours, **(e)** 3D primitives (shown in our 3D displaying software), **(f)** A 2D edge primitive, **(g)** A 2D line primitive.

- We make use of a representation of 3D contours and their relations which allows for parallel use of multiple levels of abstraction, namely accessing the original 2D information from which the 3D information is calculated as well as the local entities from which the global contours are computed.
- We address the *Local Uncertainty* problem by systematic use of the local uncertainty for making estimates on global levels. In addition, to address the *Global Uncertainty* problem, we generalize a measure for the uncertainty of the extracted 3D information from the local information to contours which gives an indication about the usability of the contour for 3D reasoning.
- 3D contours and their relations lead to a semantic description of a scene.

We demonstrate the potential of our approach in four different applications. In the context of road lane detection, we first show how attributes and relations of contours can be used to learn a Bayesian representation for lane markers in outdoor scenes and how it can be applied to lane marker detection. We demonstrate that such road structures can be characterized efficiently by contour relations. We then show how 3D contours can be used for depth prediction. In the robotic domain, we demonstrate how relations between contours can be linked to grasping actions that allow grasping of unknown objects with high success rate. Moreover, we show that through learning relevant aspects in the space of contour relations the success rate can be enhanced more, indicating (as in the lane detection case) that contours and their relations span a relevant space for learning.

The rest of the article is organized as follows. In Section 2, the early cognitive vision system that has been used for this work is introduced briefly. Then we describe the representation of the multi-modal 3D contours and their relations in Section 3, 5 and 6. Section 4 is in particular dedicated to the problem of uncertainty representation and handling. We demonstrate the relevance of our approach in four different tasks in Section 7. We conclude with a discussion in Section 8.

2. Primitives: Local Edge Descriptors

In this work, we make use of a hierarchical representation based on the early cognitive vision system presented in [32]. A calibrated stereo camera setup is used to extract along image contours sparse 2D and 3D features, called multi-modal *primitives*. A 2D primitive represents a small image patch (Figure 1) in terms of multiple visual modalities such as position $\mathbf{x}$, orientation θ, phase ϕ and color $(\mathbf{c_l}, \mathbf{c_m}, \mathbf{c_r})$. This results in a feature vector $\boldsymbol{\pi}_i = (\mathbf{x}, \theta, \phi, (\mathbf{c_l}, \mathbf{c_m}, \mathbf{c_r}))$. There are three colors associated with a primitive (left, middle and right colors respectively), since a primitive covers a patch larger than one pixel. The size of a 2D primitives' patch is derived from the bandwidth of the local filters that were used to extract them (for details see [33]). For a 3D primitive, the patch size depends on the size of the 2D primitives, propagated geometrically during during stereo reconstruction. This leads to a largely anisotropic variation in patch size, depending on the primitives' position in space (primitives close to the camera are smaller and get increasingly large with increasing distance from the camera), see [33] for a formal discussion. Note that the color information depends on whether a primitive represents a step-edge or a line structure [32] (see Figure(f-g)). As discussed in [32], this distinction can be made from phase information. For step-edges, only left and right colors are coded; for line structures, the color of the middle strip is also coded, and denoted as the 'middle' color.

2D primitives are matched across two stereo views, and pairs of corresponding 2D-primitives afford the reconstruction of a 3-dimensional equivalent called a *3D primitive*, which is encoded by the vector $\boldsymbol{\Pi}_i = (\mathbf{X}, \Theta, \Phi, (\mathbf{C_l}, \mathbf{C_m}, \mathbf{C_r}))$, where $\mathbf{X}$ is the position, Θ is the orientation, Φ is the phase and $(\mathbf{C_l}, \mathbf{C_m}, \mathbf{C_r})$ is the color of the 3D primitive.

Collinear and similar (in terms of color and phase) primitives are linked together to form structures denoted as *groups*. Groups are sets of unsorted primitives and give rise to contours

as discussed in Section 3. The resulting representation is hierarchical, with pixels at the bottom and contours at the top. Richness and reliability differ between the levels of the hierarchy: information at lower levels is more reliable; it is semantically richer at the upper levels. Therefore, different levels of the hierarchy can be used for reasoning and verification.

In Figure 1, an overview of the visual representation is presented. One stereo image pair (Figure 1(a)) is used to create the 2D features (Figure 1(c)) by using the filtering results (Figure 1(b)). These features are linked together to create 2D contours (Figure 1(d)). Two-dimensional primitives and contours are then used to find correspondences between left and right views to reconstruct 3D primitives (Figure 1(e)). Finally, 2D contours and 3D primitives are used to create 3D contours.

As discussed in [34], during the reconstruction of 3D visual entities from 2D stereo data, the uncertainty of 2D data propagates via the operations that are used for reconstruction. This uncertainty can be modeled as a covariance matrix. The uncertainty calculation of the multi-modal primitives discussed above was elaborated by Pugeault et al. [35]. For a 3D primitive $\mathbf{\Pi}_i$, the position uncertainty of $\mathbf{X}$ is modeled as a 3×3 matrix denoted as Λ_i. Geometrically, Λ_i can be interpreted as an ellipsoid (Figure 2(a)) where the orthogonal axes are the eigenvectors of Λ_i, and their lengths are given by the corresponding eigenvalues. The fact that primitives are line descriptors leads to an additional complexity in uncertainty model since the 2D edges' orientation can have a big impact on the reconstruction. This effect becomes critical when a 2D primitive's orientation approaches the epipolar line's orientation. As has been shown in [35], Λ_i is affected significantly by the right 2D-primitive's orientation (Figure 2(b)) leading to large uncertainties when the orientation gets close to $\pi/2$, the orientation of the rectified epipolar line. Moreover, the primitive's position uncertainty also vary depending on the primitive's position relatively to the cameras, in the same way as for point reconstruction (Figure 2(c-d)). Note that we do not describe the uncertainty of the orientation θ since it is not used in this paper.

3. Contours: Global Edge Descriptors

As discussed in [5], global entities such as contours provide a global overview of the scene, which makes reasoning about geometry and shape easier. In this section, we discuss 2D and 3D global entities that are built upon the multi-modal primitives presented in Section 2.

Image contours are encoded as collinear chains of primitives: if two primitives are both collinear and similar (in terms of color and phase) in an image, they are grouped together [36]. To create contours from these groups, the primitives inside the groups are sorted by their position, and the global attributes discussed in Section 5 are calculated. In the case of stereo, if the collinear and similar primitives in the left image have corresponding primitives that are collinear and similar on the right image, the reconstructed primitives are added to the same 3D group. Once the primitives contained within these groups are ordered according to their position and the attributes are calculated, they form 3D contours.

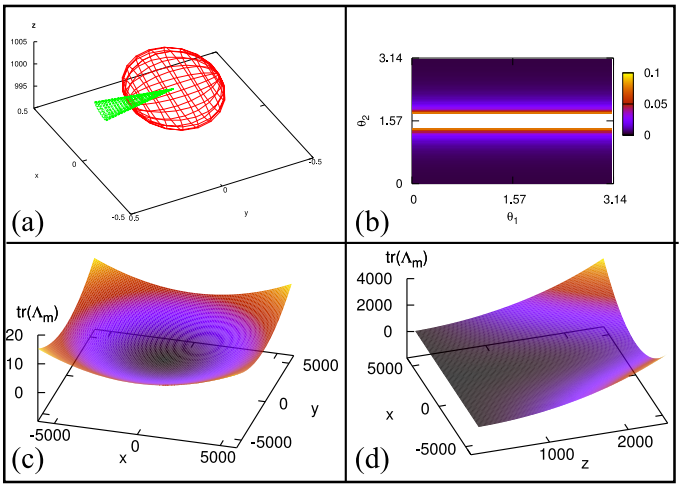

Figure 2: Position uncertainty of a 3D primitive (from [35]) (a) Illustration of position and orientation uncertainty. (b) Effect of 2D-primitives' orientations on the trace of Λ. (c) Traces of the covariance matrix Λ, for different locations $(x, y, 100)^\top$ on the xy-plane. (d) Traces of the covariance matrix Λ, for different locations $(x, 0, z)^\top$ on the xz-plane.

Linking the multi-modal primitives creates multi-modal contours that contain not only geometrical but also appearance information. Similar to primitives, contours also carry information about position, orientation and color. The details of these attributes are discussed in Section 5. These attributes together with the geometrical and visual smoothness of contours give rise to relations between contours (see Section 6) which can be used within the context of different reasoning processes.

Before we go into further details of contours and their relations, we present the notation used in this article. Most of the definitions are similar for 2D and 3D contours; therefore, the notation for 3D will be given in parentheses right after the 2D definition. We represent the i^{th} 2D contour as c_i (C_i). Every contour is composed of visual primitives, and the position, orientation and color attributes of the j^{th} primitive of the i^{th} contour are represented as ${}^p\mathbf{c}_i^j$ (${}^p\mathbf{C}_i^j$), ${}^o\mathbf{c}_i^j$ (${}^o\mathbf{C}_i^j$) and $({}^{C_L}\mathbf{c}_i^j, {}^{C_M}\mathbf{c}_i^j, {}^{C_R}\mathbf{c}_i^j)$ $(({}^{C_L}\mathbf{C}_i^j, {}^{C_M}\mathbf{C}_i^j, {}^{C_R}\mathbf{C}_i^j))$ respectively. Note that the indexes i and j are arbitrary numbers in the scope of their context and for any scene, c_i is not always the projection of C_i (Because of the matching problem during the 3D reconstruction, a 2D contour can contain the projection of more than one 3D contour).

3.1. Contour Parametrization Using NURBS

Although good continuation in terms of geometry and color provides the features that belong to the same contour as a sorted point list, having an analytical description of the contour has various advantages, including robust geometrical calculations. We make use of such an analytical description in the context of grasping (Section 7.3) where robust analytical calculations, like the tangent vector at a specific position on a curve, are necessary. In this work, we chose NURBS (Non-Uniform Rational B-Splines) [37] as a suitable mathematical framework since it is invariant under affine as well as perspective transformations, it can be handled efficiently in terms of memory and computational requirements, and it offers one common mathematical

form for both standard analytical shapes and free-form shapes. In Section 7.3, the use of NURBS in the context of grasping is presented where the orientation of a grasp is calculated as the tangent to a contour. A NURBS curve is defined as [37]

$$C(t) = \frac{\sum_{i=0}^{n} N_{i,p}(t) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(t) w_i} \quad (1)$$

where p is the order, $N_{i,p}$ are the B-spline basis functions, P_i are control points, w_i is the weight of P_i, n is the number of control points, and t is a continuous variable between 0 and 1.

Note that, because of the noisy nature of 3D data, approximating the curve gives a smoother representation than interpolation. In this work, we used a scheme similar to the one discussed in [38] to determine the 3D points that are used in approximation. For a set of points S, a random point p is chosen. Let S_i be a subset of S that contains the points that are at a distance less than r to p. After computing a regression line l for the points in S_i, the two points on l that have a distance of r to p are stored to be used in approximation and the procedure is repeated with these two points until every point on the contour is covered. We end up with a set of points that represent the curve as line segments (Figure 3(b)). We then use these points to apply a least-squares NURBS approximation (see [39] for details on approximation). A sample approximated curve is shown in Figure 3(c). In the rest of this work, we use $C_i(t)$ to denote the NURBS approximation of C_i. Note that the advantage of this approach is the reduction of the approximation points to a smaller set which is defining the skeleton of the whole set. We can than use this set to decide the number of control points in the approximation and the degree of the approximating NURBS.

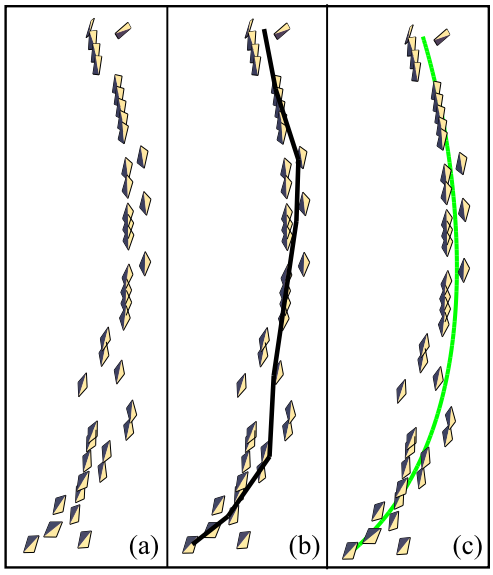

Figure 3: NURBS parametrization of a sample contour. (a) 3D primitives of a contour. (b) The line segment representation of the contour is shown with black line segments. (c) The approximated NURBS is shown as a green curve.

4. Handling the Data Uncertainty

As discussed in Section 2, the uncertainty of 2D data propagates to 3D via reconstruction and leads to geometrical noise. In Figure 4, a sample 3D contour of a scene is illustrated where local 3D entities are connected with line segments. Note that the contour shown in Figure 4(d) is the contour in Figure 4(c) from a different view-point—the noise originates from stereo reconstruction. In Figure 4(e), the dominant uncertainty axes for the primitives that are contained by the contour are shown with dashed lines where the dominant uncertainty axis is the axis of the uncertainty ellipsoid discussed in Section 2 with the highest eigenvalue.

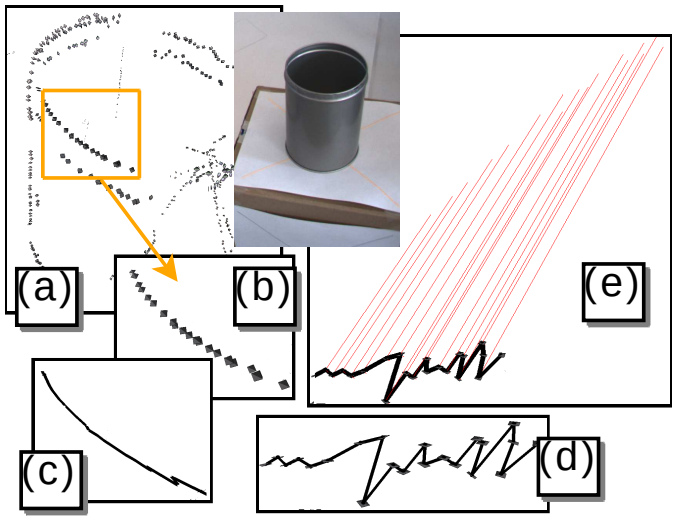

Figure 4: A sample 3D contour **(a)** 3D primitives. **(b)** 3D primitives of a selected part. **(c)** The 3D contour that contains the primitives in (b). **(d)** The contour in (c) from a different view point. **(e)** Dominant uncertainty directions.

Noise can be significant, especially for entities that are far away from the camera—as in Figure 4. When noise increases, geometrical reasoning about 3D entities becomes less stable, and 3D data becomes unusable. Although it entails extra computation, proper handling of the uncertainty leads to more stable geometrical reasoning. In this article, definitions for geometrical operations will be given with and without handling of uncertainty. In this section, a general methodology for fitting a model to a set of data points by taking the data uncertainty into account is presented (see [40, 41] for details). For such calculations, uncertainty values of the contours' primitives can be used. As discussed in Section 2, position uncertainty of a 3D primitive is modeled as a covariance matrix which defines an ellipsoid in 3D, where the eigenvectors of the matrix represent the directions of the orthogonal axes and the eigenvalues of the matrix represent the length of each axis. The parameters of a model (e.g., the parameters of a plane) are be calculated by minimizing the Mahalanobis distance between each primitive and the closest point on the model. Given a set of 3D primitives $\{\mathbf{\Pi}_0 ... \mathbf{\Pi}_n\}$, the error to be minimized for $\mathbf{\Pi}_i$ is given by:

$$e_i^2 = (X_i - \hat{X}_i)^T \Lambda_i^{-1} (X_i - \hat{X}_i) \quad (2)$$

where X_i is the position of $\mathbf{\Pi}_i$, $\hat{X}_i$ is the closest point on the model, and Λ_i is uncertainty matrix of $\mathbf{\Pi}_i$. As discussed in [41], since the uncertainty matrices of the data are unrelated, the minimization problem cannot be solved in closed form. Also, $\hat{X}_i$ is not the closest point on the model in Euclidean sense but it is the point that minimizes Eq. 2. Therefore, $\hat{X}_i$ is determined as the closest point on the model after a whitening transform [42, 40] is applied to both X_i and the model. Note that the transformation that is required to transform an arbitrary point Y into the whitened space defined by (X, Λ) is

$$\hat{Y} = WE(Y - X) \tag{3}$$

where E is the matrix that is defined by the eigenvectors of Λ, and W is the diagonal matrix defined by $1/\sqrt{e_i}$ for the eigenvalues e_i of Λ. Once the model is whitened, the closest point in the model is found and unwhitened with the following formula

$$Y = E^{-1}U\hat{Y} + X \tag{4}$$

where U is the diagonal matrix defined by $\sqrt{e_i}$. The minimization of Eq. 2 is done by the Levenberg-Marquardt method. Note that to fit a model with M parameters to N data points, an NxM Jacobian matrix (J) is required, where

$$J_{ij} = \frac{\partial e_i}{\partial h_j} \tag{5}$$

Here, e_i is the error calculated with Eq. 2 for the parameter set h. This partial derivative can be calculated numerically as

$$\frac{\partial e_i}{\partial h_j} = \frac{e_i(h) - e_i(\hat{h})}{\epsilon} \tag{6}$$

where $\hat{h}_j = h_j + \epsilon$ and ϵ is a small positive real number. A detailed explanation of the procedure for plane fitting can be found in [40].

Uncertainty information is necessary not only for geometrical calculations but also to obtain an overall uncertainty value for the whole contour, which then can be used to judge the usability of the contour for reasoning. In this work, the uncertainty of a contour is calculated as the mean of the traces of the uncertainty matrices of the primitives that are part of the contour, and is denoted as

$$\Lambda_i^C = \frac{1}{N}\sum_{k=0}^{N} \mathrm{tr}(\Lambda_k) \tag{7}$$

where Λ_k is the uncertainty matrix of the k_{th} primitive in Λ_i^C and the trace is equal to the sum of eigenvalues of the matrix, which corresponds in our case to the sum of the lengths of the uncertainty axes of a primitive. Note that this measure is used for determining the usability of 3D contours for reasoning but does not model the contour's uncertainty geometrically. Therefore, all geometrical computations use the local features' full covariance matrices rather than the contours' uncertainties.

In this section, we discussed an iterative method for model fitting to data with uncertainty. Note that for time-critical applications, an iterative solution may put a big overhead on the system. One can neglect the effect of data uncertainty in Eq. 2 by setting Λ_i^{-1} to the identity and reducing the problem to minimization of the Euclidean distance between primitives and their closest point in the model. In this case, a whitening operation is not necessary, and specific models like planes and lines can be fitted with closed-form solutions. Therefore, for the geometrical definitions in Section 5 and 6 where a plane or a line fitting is necessary without data uncertainty handling, Principal Component Analysis (PCA) is used for model fitting. For plane fitting, the eigenvector of the smallest eigenvalue in PCA provides the normal of the plane. For line fitting, the eigenvector of the highest eigenvalue in PCA provides the orientation of the line.

5. Contour Attributes

Since contours are created from multi-modal primitives based on good continuation in terms of geometry and color, they also carry geometrical and appearance information. The contours' multi-modality originates in the local entities they are computed from. Therefore, every contour has some attributes such as mean color (with left, middle and right components), position and orientation. While mean color encodes a contour's appearance, position and orientation describe its geometrical properties.

Orientation:. The orientation of a contour (both in 2D and 3D) is defined as the dominant direction of the contour, and is represented as $A^o(c_i)$ ($A^o(C_i)$). In 2D, this direction can be defined as the first principal component of the contour. In 3D, if the uncertainty of the data is not taken into account, the principal components of the contour can also be used as the dominant direction. To take the uncertainty into account, this definition can be extended to the orientation of the line that minimizes Eq. 2. The general procedure that was discussed in Section 4 can be used to solve the minimization problem, and initializing the model with the principal components of the contour reduces the amount of iterations needed for convergence. In Figure 5(a), the orientation of a sample contour is illustrated.

Position:. The position of a contour $(A^p(c_i), A^p(C_i))$ is defined as the projection of the contour primitives' centroid onto the best-fitting line (see Figure 5(a)). As discussed for the orientation, the calculation of the best-fitting line may change, and the position of a contour depends on this calculation.

Mean Color:. The good continuation criterion applied to the color modality implies that all primitives that create the contour share a similar color. Since primitives have left, right and middle colors, every contour has mean left ($A^{\vec{c}_L}(c_i)$), right ($A^{\vec{c}_R}(c_i)$) and middle ($A^{\vec{c}_M}(c_i)$) colors as well. Also, since color is an appearance attribute, it is calculated in 2D for both 2D and 3D contours. For 3D contours, the 2D contours can be obtained by projecting the 3D contours on to the image plane.

The left and right sides of a primitive with position x and orientation θ is determined according to the vector that passes through x with orientation θ. In some cases, this assignment

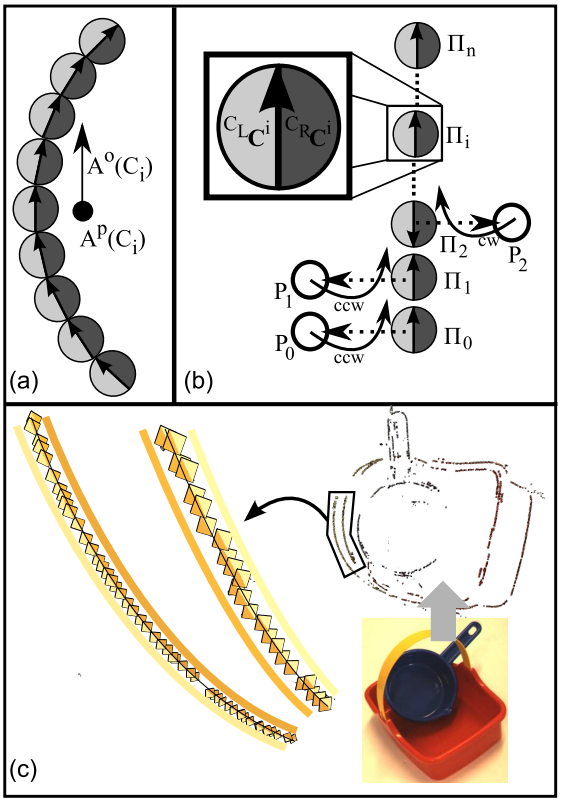

Figure 5: Illustration of contour attributes. **(a)** Orientation and position of a sample contour. **(b)** Calculation of mean color for a contour. **(c)** Mean left and mean right colors of two 3D sample contours.

may be ambiguous since edges with orientations θ and $\theta + \pi$ have the same primitive orientation. In computing mean color, this ambiguity is resolved by the following algorithm:

Given a 2D primitive $\boldsymbol{\pi}_k$ with orientation (n_i, n_j), the left color is defined as the color of the patch which is in the direction of $(-n_j, n_i)$. We use this fact to calculate the mean left and right colors. For the first primitive in the contour, we calculate a point in the left-perpendicular direction of the primitive $(-n_j, n_i)$ and check whether the movement from this point to the primitive itself and the next primitive (e.g., the movement from P_0 to $\boldsymbol{\Pi}_0$ and $\boldsymbol{\Pi}_1$ in Figure 5(b)) is counterclockwise or clockwise. For the rest of the primitives in the contour, the same procedure is applied and for the movements in the same direction of the first movement (e.g., $P_1\boldsymbol{\Pi}_1\boldsymbol{\Pi}_2$), the left color of the primitive is used for the mean left color of the contour. The algorithm is illustrated in Figure 5(b) for a sample contour. The mean color is defined in the CLab color space because of the statistically less correlated behavior [43] of the CLab space. Also, CLab is more appropriate for defining numerical differences between two colors, since the perceived difference between two colors and their Euclidean distance has a low correlation in RGB space [43]. In Figure 5(c), left and right mean colors of two sample contours are illustrated.

6. Second-Order Contour Relations

In Section 5, a couple of geometrical and appearance-based attributes were defined. These attributes together with the geometrical and visual smoothness of contours give rise to relations between contours that can be used within the context of various reasoning processes (see Section 7). In the present section, we describe certain contour relations with and without data uncertainty handling, and evaluate their potential and limits for reasoning. Note that the geometrical relations are defined only for 3D contours, since they are not preserved by perspective projection.

Angle:. The angle between two contours is defined by using the orientations of the contours as:

$$R^A(C_i, C_j) = acos\left(\frac{A^o(C_i) \cdot A^o(C_j)}{|A^o(C_i)|\,|A^o(C_j)|}\right) \tag{8}$$

Note that the definition of contour orientation may include data uncertainty handling; this will be reflected automatically in the definition of an angle. The procedure is illustrated in Figure 6(a).

Normal Distance:. The normal distance between two contours is defined by the distance from one contour's position to to the line created by the other's orientation and position (see Figure 6(b)). Therefore, the distance between the i^{th} and j^{th} contours in the scene is defined as:

$$R^D(C_i, C_j) = \frac{|\mathbf{w}_i - (\mathbf{w}_i \cdot \mathbf{u}_i)\mathbf{u}_i| + |\mathbf{w}_j - (\mathbf{w}_j \cdot \mathbf{u}_j)\mathbf{u}_j|}{2}, \tag{9}$$

where $\mathbf{w}_i$ is $(A^p(C_j) - A^p(C_i))$, $\mathbf{w}_j$ is $(A^p(C_i) - A^p(C_j))$, and $\mathbf{u}_i$ is the orientation of the i^{th} contour. Note that both terms of the sum in Eq. 9 are the formula for the distance of a point to a line.

Coplanarity:. The coplanarity of entities can be measured by their elongation with a common plane. We define the coplanarity between two contours as the mean angle between a common plane and the best-fit lines of the contours (see Figure 6(c)). Therefore, the coplanarity between the i^{th} and j^{th} contours in the scene is denoted as $R^P(C_i, C_j)$ and defined as

$$R^P(C_i, C_j) = \frac{1}{2}\left(\pi - acos\left(\frac{\mathbf{n} \cdot A^o(C_i)}{|\mathbf{n}||A^o(C_i)|}\right) - acos\left(\frac{\mathbf{n} \cdot A^o(C_j)}{|\mathbf{n}||A^o(C_j)|}\right)\right), \tag{10}$$

where $\mathbf{n}$ is the normal of the common plane $\Psi(C_i, C_j)$ defined by the primitives inside the contours. Similar to other geometrical definitions, data uncertainty can be handled in the process of common plane calculation and finding the best fitting line (see Section 4 for details).

Cocolority:. The cocolority $R^{CL}(C_i, C_j)$ between two contours is defined as the color difference between the mean colors on the contours' sides that face each other. The color difference is calculated by the CIE 1994 color difference [44]. A mean color is calculated for the contours' sides that face each other, and the color difference is then used to estimate how reliable the mean color is.

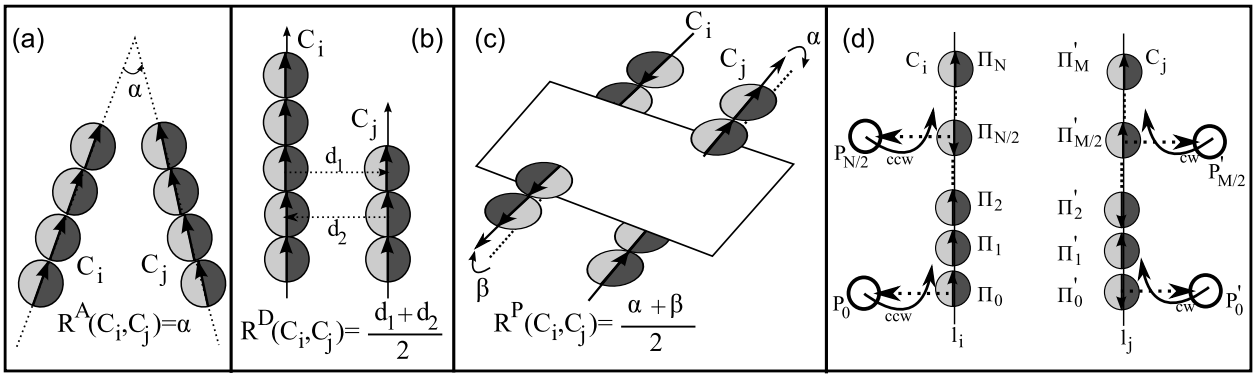

Figure 6: Illustration of contour relations. **(a)** Angle **(b)** Normal distance. **(c)** Coplanarity **(d)** Cocolority.

To find the facing sides of two contours, an algorithm similar to the mean color calculation is used. The procedure is illustrated in Figure 6(d). For 3D contours C_i and C_j, we first calculate two points (P_0,P'_0) that are on the left side of the first primitive ($\mathbf{\Pi}_0$,$\mathbf{\Pi}'_0$) of each contour. We then calculate two more points ($P_{N/2}$,$P'_{M/2}$) that are on the same side with P_0 and P'_0. Note that $P_{N/2}$ and $P'_{M/2}$ are calculated from the primitives that are closest to the contours' centroid ($\mathbf{\Pi}_{N/2}$ and $\mathbf{\Pi}_{M/2}$. A side test to P_0 (P'_0) with respect to the line defined by the other contour reveals which side of the contour faces the other one.

Similar relations can be defined for the local features discussed in Section 2. Within the context of Section 7, relations between primitives are used for comparison between local and global reasoning approaches. A brief explanation of these relations can be found in [45].

6.1. Effect of Contour Uncertainty on Relations

To investigate the effect of uncertainty on the contour relations defined in Section 6, an artificial dataset was produced where an isosceles triangle is moved away from the camera in a direction perpendicular to the image plane. As illustrated in Figure 7(a-b), as the triangle moves further from the camera, the uncertainty of the contours becomes higher and the poor reconstruction makes reasoning more and more difficult.

In the experiment, angle and coplanarity relations were calculated for contours that lie on the isosceles sides of the triangle in each frame. For every frame, the number of primitives contained within a contour was kept constant. The experiment was repeated 100 times, adding random white noise on the stereo image pair. The resulting plots are presented in Figure 7(c-d). The correct 3D angle is 30 degrees, and the coplanarity value is zero (since both contours are in the same plane).

Two aspects of these results deserve a detailed explanation.

- **Degeneration:** The precision of the estimates decreases with triangle's distance from the cameras. This degeneration process is reduced if using estimates that take the uncertainties into account; however, it cannot be avoided. At some point, it does not make sense anymore to do such 3D reasoning processes.
- **Systematic errors:** Not only the variance of the estimates increases with distance, there is also a growing systematic error in these estimates. The reason for this is that isotropic noise on the orientation of the two contours results in a bias. It is more likely that these orientations deviate rather than converge for an angle of 30 degrees.

Any reasoning with 3D contour relations is affected by these two hindrances.

7. Using 3D Contours in Cognitive Vision and Grasping Tasks

In this section, we exemplify the use of our multi-modal contours and their relations, with four applications from different domains. We chose these very different domains—road lane detection, depth prediction and grasping unknown objects—to demonstrate the generality of the approach. In Section 7.1, the contour relations are used in a Bayesian framework for lane marker detection. In Section 7.2, we show that 3D contours can be used for depth prediction, even for non-planar homogeneous surfaces. In Section 7.3, we demonstrate that 3D contours and their relations can be used to define grasping hypotheses for unknown objects, and that the very same relational space can be used as features for grasp learning. Finally, in Section 7.4, we discuss how the effect of uncertainty can be reduced via accumulation over time.

7.1. Task 1: Road Lane Detection

Relations between contours give rise to both geometrical and appearance cues. These cues can be combined within a Bayesian framework to calculate the probability of an event happening for given cues. In this section, a road lane detection system based on Bayesian reasoning (see, e.g., [46]) is discussed to illustrate the relevance of the relational space for reasoning. The details of this work can be found in the workshop publication [45].

Lane marker detection usually combines different stages of processing such as feature extraction, initial estimation, lane modeling and tracking (see, e.g., [47, 48] for a review). In this section, we focus on the estimation and modeling steps. In 2D approaches such as [49, 48] perspective properties of parallel lines and appearance properties of lane markers are used to

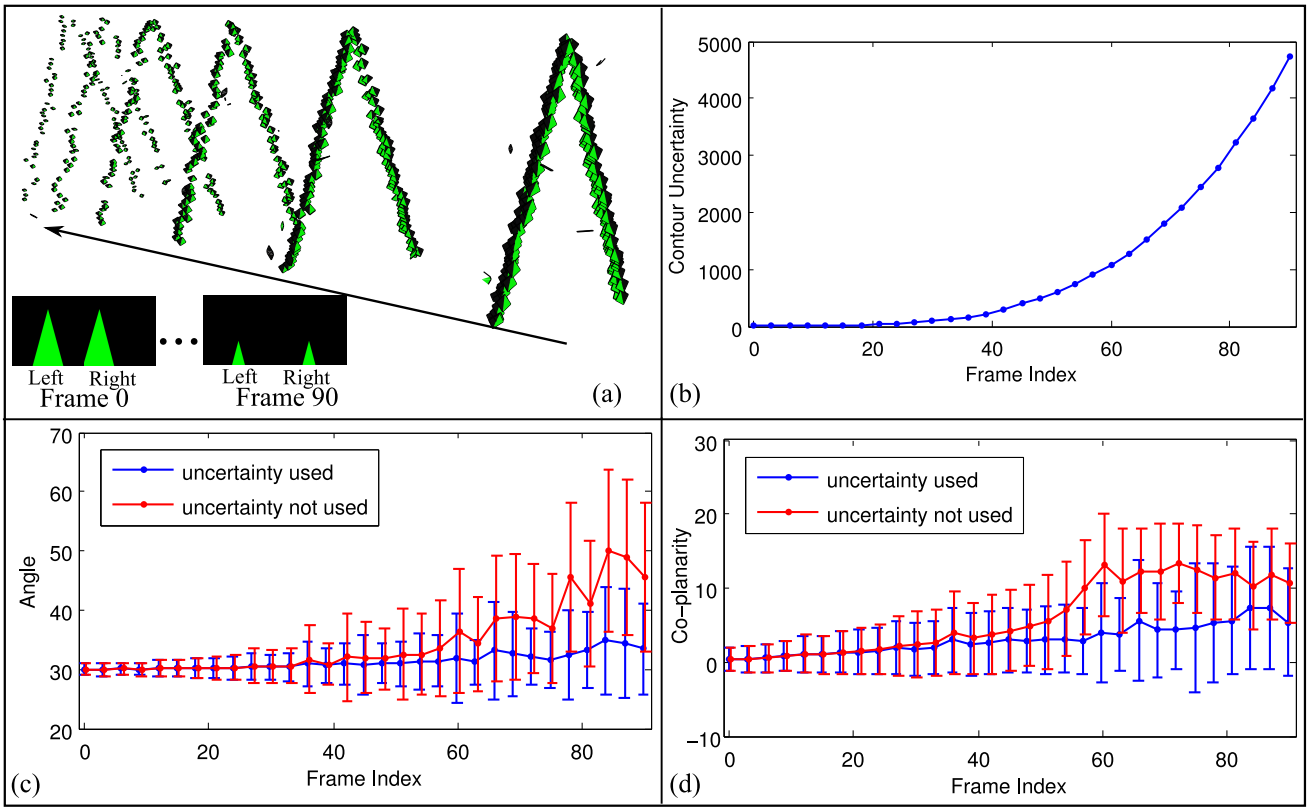

Figure 7: Effect of uncertainty on contour relations for the isosceles sides of the triangle in an artificial dataset. (a) Some samples from the dataset as stereo image pairs and 3D primitives. (b) Change in the contour uncertainty as the triangle moves away from the camera. (c) The angle between the sides of the triangle as it moves away from the camera. (d) The coplanarity between the sides of the triangle as it moves away from the camera.

model the lanes. 3D approaches such as [50] combine geometrical properties of roads (e.g., the width of the road, estimated ground plane) and appearance properties in a statistical manner. In our approach, we start with the observation that lane structures are defined not only by their local appearance but also by the relations between them (see Figure 8): lane structures, besides usually being connected to collinear, high-contrast edges, are usually found close to the ground plane, are mutually coplanar, and are part of parallel structures with certain distance ranges relative to the lane width. None of these different relations is on its own sufficient to describe lane structures; however, their probabilistic combination in terms of a Bayesian reasoning process results in a stable classification. Moreover, we do not assume a static road model but learn it in terms of prior and conditional probabilities.

In addition to the angle, normal distance and coplanarity relations discussed in Section 6, we use color difference between mean left and right colors of a contour and the Euclidean distance of contours to a roughly estimated ground plane (ground plane relation). Note that a rough estimation of the ground plane is possible, once the height of the camera as well as its orientation is known.

The likelihood of an entity e being part of a lane $\mathcal{L}$ given a number of visual cues pertaining to the entity $\eta^e = \{\eta_0^e, ..., \eta_n^e\}$ can be defined as:

$$P(e \in \mathcal{L}|\eta^e). \tag{11}$$

According to Bayes' formula, Eq. 11 can be expanded to

$$\frac{P(\eta^e|e \in \mathcal{L})P(e \in \mathcal{L})}{P(\eta^e|e \in \mathcal{L})P(e \in \mathcal{L}) + P(\eta^e|e \notin \mathcal{L})P(e \notin \mathcal{L})}. \tag{12}$$

Assumming independence between the cues, the denominator of Eq. 12 can be expanded to

$$P(\eta_1^e, \ldots, \eta_n^e|e \in \mathcal{L}) = P(\eta_1^e|e \in \mathcal{L}) \cdot \ldots \cdot P(\eta_n^e|e \in \mathcal{L}) \tag{13}$$

and

$$P(\eta_1^e, \ldots, \eta_n^e|e \notin \mathcal{L}) = P(\eta_1^e|e \notin \mathcal{L}) \cdot \ldots \cdot P(\eta_n^e|e \notin \mathcal{L}). \tag{14}$$

Eq. 12 can be solved once the prior and conditional probabilities are known. In the learning phase, a set of hand-labeled data from a publicly-available sequence (`www.mi.auckland.ac.nz/EISATS`, SET 3: Colour stereo sequences Drivsco) was used to calculate these probabilities for the training set. The prior probability $P(e \in \mathcal{L})$ is calculated by counting the entities that are on and off the lane, for all training images (see Figure 9(a)). Also, in order to calculate the conditional probabilities $P(\eta^e|e \in \mathcal{L})$ and $P(\eta^e|e \notin \mathcal{L})$ for a specific value of η_i^e in the test phase, a conditional probability density is estimated from the training set for each relation (Figure 9(b-f)). Note that for a given conditional probability, the non-overlapping area between two densities can be expressed as an $L^1 - distance$, which can be associated to the relevance of each relation for reasoning. As illustrated in Figure 9, the L^1-distance is quite high for

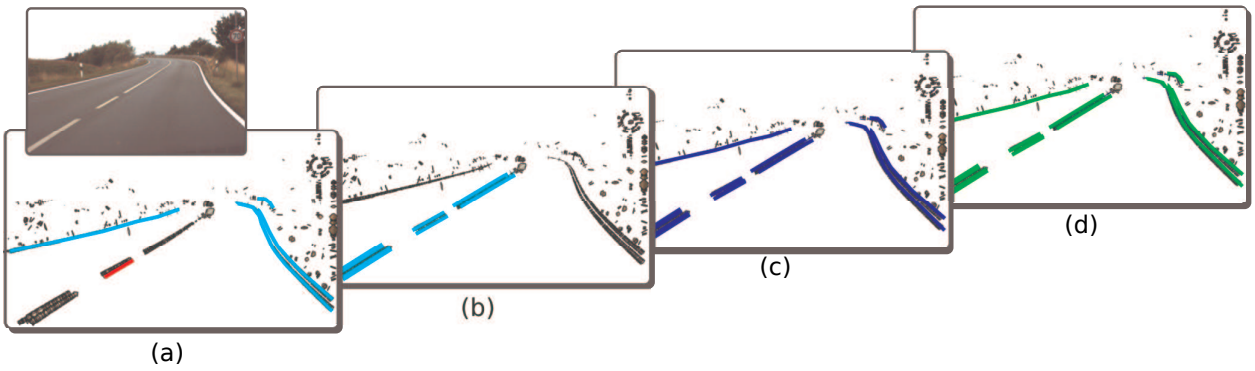

Figure 8: Illustration of 3D relational cues. (A selected 3D contour is marked in red in **(a)**). **(a)** All contours that have 1500–3500 mm normal distance to the selected contour. **(b)** All contours that have 0–200 mm normal distance to the selected contour. **(c)** All contours that are coplanar to the selected contour. **(d)** All contours that have 0–5°angle to the selected contour.

some cues (e.g., the distance to the ground plane), which is an indication of that particular cue's relative importance. In this work, we represent the probability densities with histograms, and for the densities X and Y with n bins, the L^1-distance is defined as:

$$\|X - Y\|_1 = \sum_{i=0}^{n} |X_i - Y_i|. \tag{15}$$

Once the prior probabilities and the densities for the conditional probabilities are calculated from the training data, Eq. 12 can be used to calculate the posterior probabilities—i.e., the probability of entities of the test set being on the lane for the given visual cues. In this work, the Bayesian framework was applied to three cases where the relational space was based on (a) relations between primitives, (b) relations between contours, and (c) relations between contours that have an uncertainty value below a certain threshold. In Figure 10, the distribution of the posterior probabilities for all three cases are displayed. Note that the L^1-distance value is highest for the case where contours are used after an uncertainty thresholding. In Figure 11, some samples of extracted lanes are shown.

The evaluation was done by measuring two values for each case. We calculate the classification success rate (CSR) as the percentage of true positives (entities that are labeled as part of a lane and detected as part of a lane) plus true negatives (entities that are labeled as not part of a lane and detected as not part of a lane) in the whole set (Eq. 16). We also have a positive success rate (PSR), which is defined in Eq. 17 as the percentage of true positives in the set of true positives plus false negatives (entities that are labeled as part of lane and detected as not part of lane).

$$CRS = \frac{true\ positives\ +\ true\ negatives}{whole\ set} \tag{16}$$

$$PSR = \frac{true\ positives}{true\ positives\ +\ false\ negatives} \tag{17}$$

While the classification success rate measures how successful the algorithm is at classifying entities in a scene as lane and non-lane, the positive success rate measures how successful the algorithm is for finding lane structures. When the relations between primitives are used, we obtain a classification success rate of 78.4% and a positive success rate of 58%. The use of contour relations results in a classification success rate of 75.3% and a positive success rate of 74.2%. Once the contour relations are used after applying an uncertainty threshold, we obtain a classification success rate and a positive success rate of 88.3%.

7.2. Task 2: Depth Prediction using Contours

In this subsection, we show that depth information at homogeneous image areas can be predicted from the depth information available at the contours (an extension of the work originally presented in [51]). It is not possible to estimate depth information at homogeneous image areas directly. The statistical investigations using chromatic images with corresponding range data have revealed that it is possible to predict the depth information at homogeneous image areas from the reliable depth information available at the contours of a scene [52].

Similar to an edge primitive, we define a homogeneous primitive in 2D (π^m) and in 3D (Π^m):

$$\pi^m = (\mathbf{x}, \mathbf{c}), \tag{18}$$

$$\Pi^m = (\mathbf{X}, \mathbf{n}, \mathbf{c}), \tag{19}$$

where $\mathbf{x}$ and $\mathbf{X}$ are the positions in the 2D and the 3D space, respectively; $\mathbf{c}$ is the color representation, and $\mathbf{n}$ is the surface normal of the homogeneous primitive in the 3D space. π^m is extracted directly from the image (see Figure 12(b)), and Π^m is estimated from the contours that bound π^m.

The bounding contours of a homogeneous primitive π^m are found by making searches in a set of directions d_i, $i = 1, ..., N_d$ for the edge primitives. In each direction d_i, starting from a minimum distance R_{min}, the search is performed up to a distance of R_{max} in discrete steps s_j, $j = 1, ..., N_s$. If an edge primitive π is found in direction d_i in the neighborhood Ω of a step s_j, π is added to the list of bounding edges and the search continues with the next direction d_{i+1}.

The bounding contours of a homogeneous primitive π^m form a set of edge primitives, $\{\pi_i\}$ (for $i = 1, ..., N_E$). We form pairs from this set (only from those that are cocolor, coplanar and not collinear), and each pair gets to cast a vote $\mathbf{v}$ for the depth of

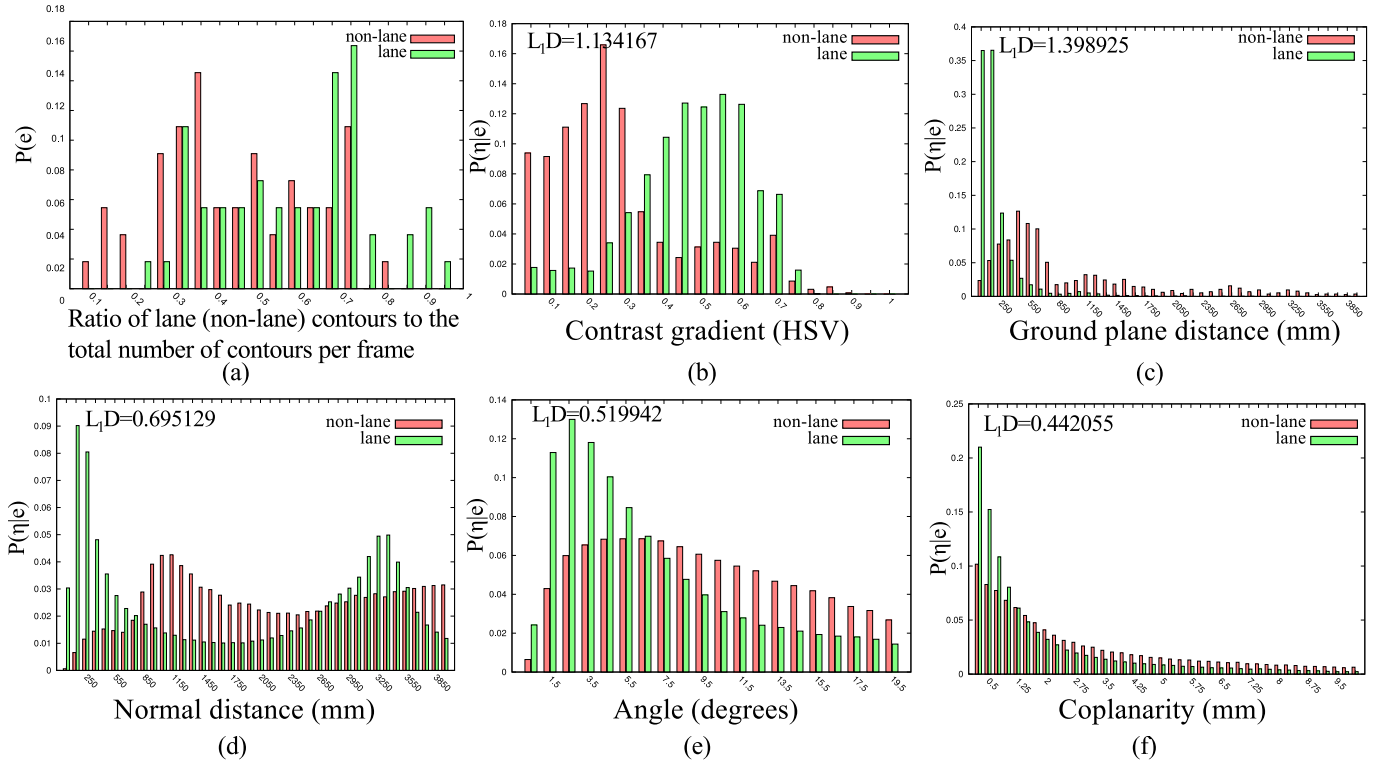

Figure 9: The prior probability distribution (a) and conditional probability densities (b-f) for relations between contours that contain a maximum of 6 primitives.

π^m, which is defined as

$$\mathbf{v} = (\mathbf{X}, \mathbf{n}). \tag{20}$$

We compute the vote from a pair of edge primitives π_i and π_j by first fitting a plane to their corresponding 3D primitives Π_i and Π_j, and finding the intersection of this plane with the ray going through the homogeneous primitive π^m and the optical center of the camera. This intersection defines the 3D position $\mathbf{X}$ in the vote $\mathbf{v}$, and the surface normal of the plane at point $\mathbf{X}$ is taken as $\mathbf{n}$.

Each vote has an associated reliability r, which is inversely proportional to the distance between the homogeneous primitive π^m and the voting pair of edge primitives π_i and π_j:

$$r_i = 1 - \frac{1}{min(d(\pi^m, \pi_i), d(\pi^m, \pi_j))}, \tag{21}$$

where $d(.,.)$ is the Euclidean distance between two features.

The set of votes is combined by, first, clustering the votes (using a histogram with a fixed number of bins) and averaging (using the reliability in Eq. 21 as the weights) the votes inside the most crowded cluster.

The model described above which focuses on local features can only predict depth for planar surfaces. This limitation can be relaxed by making use of the global contours. We create a one-to-one association between the primitives of curved contours that surround a homogeneous area and do the depth prediction based on the paired primitives (see [52] for details). The model's applicability to round surfaces is demonstrated on a simple scene in Figure 12. The figure also shows the results from two dense stereo algorithms; namely, dynamic programming and a phase-based stereo algorithm [53]. In addition, the applicability of the model is presented on an outdoor scene in Figure 12(j-l). Comparing the results in Figure 12(k) and Figure 12(l), it is visible that the depth prediction gives reasonable information for areas sufficiently close to the camera, however the quality of depth prediction degrades when the reconstruction quality of contours decreases with increasing distance from the camera. Here we want to stress that we do not claim that our method outperforms other stereo methods which are based on direct correspondences. We want to show however that depth prediction based on contours provide one additional cue that can be combined with other depth cues. For example, in [54], this depth estimate was used to improve dynamic programming stereo-matching.

7.3. Task 3: Using 3D Contours and Their Relations for Grasping Unknown Objects

In this section, the use of 3D contours and their relations in the context of grasping unknown objects is presented. We demonstrate that 3D contours and their relations can directly be associated to actions and that they span a relevant space as features for learning.

In the absence of prior knowledge about the 3D model of an object, sensory data must be used to calculate grasping hypotheses (see, e.g., [55]). The sensory data is analyzed to identify few grasping points that are likely to be grasped and the outcome of the grasping attempts are analyzed further to generalize the grasping behavior (see, e.g., [56, 57, 58]). For example in [56],

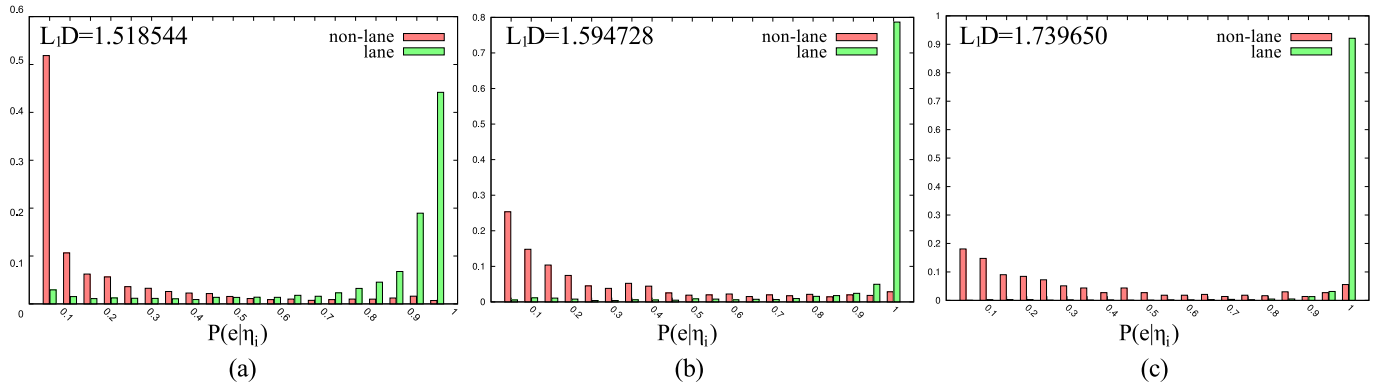

Figure 10: The normalized histograms of the posterior probability densities for individual and grouped entities. (a) Based on individual primitives. (b) Based on groups of maximum 6 primitives without uncertainties. (c) Based on groups of maximum 6 primitives after applying an uncertainty threshold.

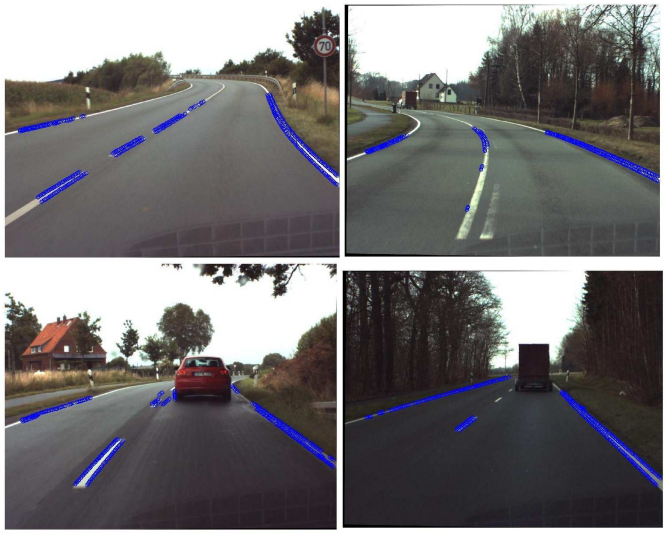

Figure 11: Results of the Bayesian framework applied on different frames.

Saxena et al. uses multiple images of the object to be grasped and grasping locations that are identified on these images are triangulated for gathering 3D information. Similar to our approach, 3D information coming from different sensors (such as stereo cameras or range scanners) can be used directly to generate grasping hypotheses. For example in [58], 3D data points are clustered into primitive box shapes to approximate unknown objects and grasping hypotheses are created based on this box representation. The main difficulty of grasping of unknown objects is that features extracted from the scene (which in general are afflicted with a large degree of noise) need to be related to grasping actions. Here we show that the 3D multi-modal contours can be a powerful trigger for such grasping actions. In particular, we show that a more global approach based on contours outperforms a local approach based on local primitives in terms of stability and robustness.

Grasps for a two-finger gripper are defined through a 3D location x and two direction vectors r_1 and r_2 as shown in Figure 13(a). Making use of the coplanarity relation as defined in Section 6, we can associate a number of grasps to two coplanar contours or primitives (see Figure 13(b)). In this way, we can compute a large variety of grasping hypotheses (see Figure 13(d)) for any given object.

A grasp at a certain position $C_i(t_0)$ on a contour C_i that is coplanar and cocolor with a contour C_j is defined as:

$$x = C_i(t_0), \quad r_2 = n, \quad r_1 = r_2 \times C_i'(t_0) \tag{22}$$

where $C_i(t_0)$ is the 3D position on the NURBS curve of C_i (see Section 3.1), $\mathbf{n}$ is the normal of the common plane $\Psi(C_i, C_j)$ that is calculated for coplanarity, $C_i'(t_0)$ is the first derivative of $C_i(t_0)$ which corresponds to the tangent vector of the curve at that position and $\times$ is the cross product operator. An example grasp defined by two contours is illustrated in Figure 13(e). It has been shown that such a straightforward association of coplanar contours to grasps, without making use of any prior object knowledge, is already capable of achieving success rates of about 40% even for scenes with high complexity such as the one in Figure 14 (for details, see [59]).

In the following, we compare the performance of grasps defined by relations between contours and local features. For the local approach, the grasping hypotheses are calculated from the location and orientation of two primitives that are cocolor and coplanar as described in [55] (see Figure 13(b) for a brief explanation).

For the comparison of the global and local approaches, two contours from the brim of the can that is shown in Figure 15 were manually selected to make sure that they belong to the same surface so that the orientation of the brim can be used as ground truth. After adding random noise to all primitives within the range of ellipsoids defined by their position uncertainty (see Section 2), grasp hypotheses were calculated using both the local and global approaches for a number of locations on the brim of the can. For the local approach, every third primitive on one contour and all other primitives on the other contour were used to define multiple grasping hypotheses and the hypothesis that aligns best with the ground truth was chosen. For the global approach, grasping hypotheses were defined using Eq. 22 for the locations used for the local approach. As shown in Figure 15 (a) and (b), even though the best grasps were chosen for the local approach, the global approach performs significantly better.

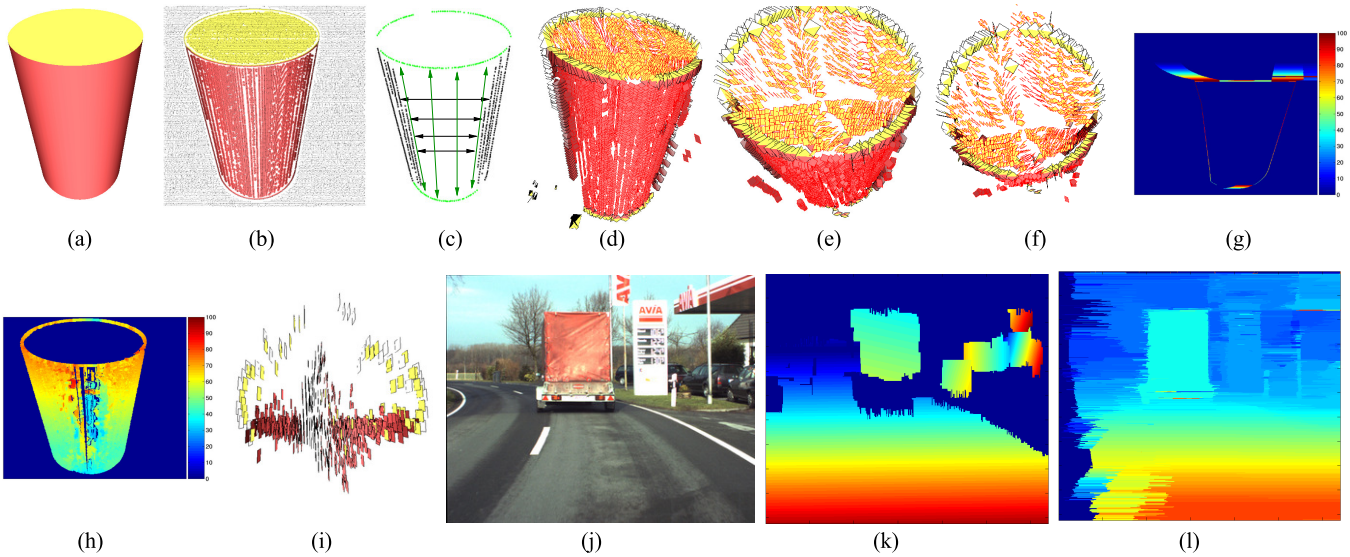

Figure 12: Experiment results on a cylinder. **(a)** Left image of the a stereo pair. **(b)** Edge and homogeneous primitives extracted from (a). **(c)** Contours. Green shows the curved contours. **(d-f)** The predictions of our model shown as snapshots from our 3D displaying software. As surface fitting for curved surfaces in case of outliers is not trivial, we are unable to provide disparity maps for our results. **(g)** Disparity map using Dynamic Programing. **(h)** Disparity map computed using [53]. **(i)** Top view of (h). **(j)** A real outdoor scene, and the result of depth prediction as a disparity map **(k)**. **(l)** Result of Dynamic Programming on the road scene.

This is because the local approach is based on only two primitives, while the global approach makes use of the geometric stability of contours.

Besides the fact that using coplanar contours instead of coplanar pairs of local features increases the robustness and success rate of grasping unknown objects, it has been shown that the performance can be further increased by learning [59]. To this end, we constructed an artificial neural net that predicts a success likelihood for a grasp depending on the relations of the contours defining the grasp. Three relations between contours in addition to coplanarity—namely, cocolority, normal distance, and angle—were used as input features to the neural network. Each grasp attempt was evaluated haptically by the robot, which resulted in a set of triplets containing the performed grasp, the contours defining the grasp as well as a label for success or failure. From these data, the robot system learned a function predicting the success likelihood of the grasp depending on the values of the four relations. For example, from the large number of potential grasps shown in Figure 13(d), the system picked the one with the highest predicted success likelihood. We have shown that such a learning based selection can increase the success rate of the grasping behavior from about 40% to above 60% (for details, see [59]). Note that by learning a success likelihood for a grasp that can be associated to a certain constellation of contours, no prior knowledge pertaining to specific objects is introduced. Rather, the system acquires general knowledge about the chance of grasp success when certain sets of relations occur in the scene.

7.4. Task 4: Disambiguation via Accumulation

As we discussed in Section 6.1, data uncertainty has a negative effect while reasoning in 3D. One way to reduce the uncertainty is to acquire multiple images of a scene from different perspectives to accumulate a full 3D model [60]. This process is based on the combination of three components. First, all primitives are tracked over time and filtered using an Unscented Kalman Filter based on the combination of prediction, observation and update stages. In the prediction stage, the system's knowledge of the motion (e.g., the motion of the robot arm) or an estimated motion (see, e.g., [61]) is used to calculate the poses of all accumulated primitives at the next time step. The observation stage matches the predicted primitives with their newly observed counterparts. The update stage corrects the accumulated primitives according to the associated observations. This allows the encoding and update of the feature vector. Secondly, the confidence in each tracked primitive is updated at each time step according to how precisely the accumulated primitive was matched with a new observation. The third process takes care of preserving primitives once their confidences exceed a threshold, even if they later become occluded for a long period of time. It also ensures that primitives are discarded if their confidence falls below a threshold. New primitives that were not associated to any accumulated primitive are added to the accumulated representation, allowing the progressive construction of a full 3D model.

Temporal consistency is used by several computer vision methods as well to reduce feature uncertainty, prominently the Structure From Motion (SFM) and Simultaneous Localization and Mapping (SLAM) class of methods. SFM methods are mainly batch methods (e.g., bundle adjustement [62]) that can achieve very high accuracy by minimizing the projection error of the 3D features in all images. In the SLAM scenario, in contrast to the bundle adjustment case, the maps are built incrementally using Kalman filter variants [63, 64, 65, 66, 67] or, more recently, particle filters [68, 69]. The Fly algorithm by [70] that uses genetic algorithms to maintain a population of "flies" on

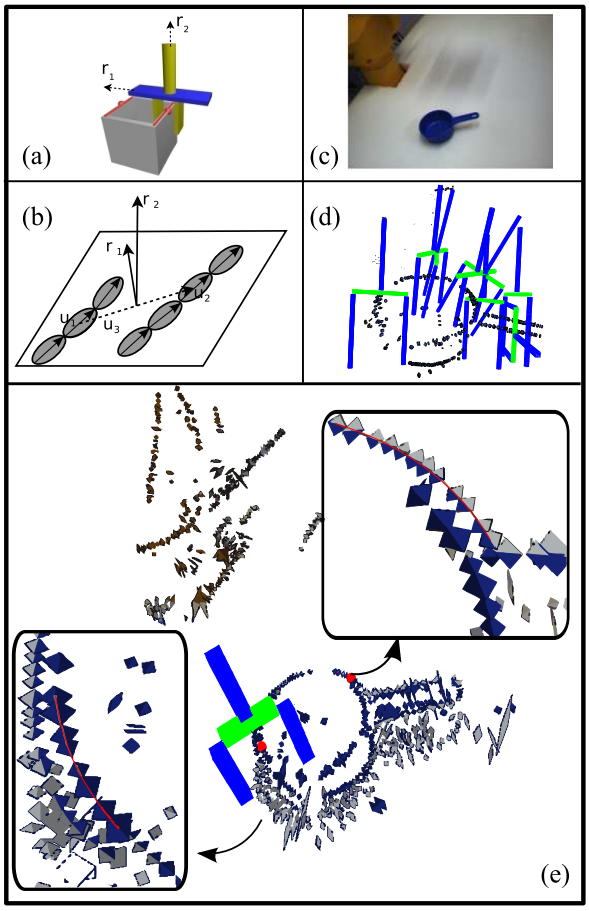

Figure 13: Grasping with a two-fingered gripper. **(a)** A grasp is defined by a location and two directions (r_1,r_2). **(b)** The direction vectors for local features are defined as $r_2 = (u_3 \times u_1) + (u_2 \times u_3)$ and $r_1 = u_3 \times r_2$. **(c)** A blue pan to be grasped. **(d)** A set of grasping hypotheses generated to grasp the blue pan. **(e)** An example grasp based on contours.

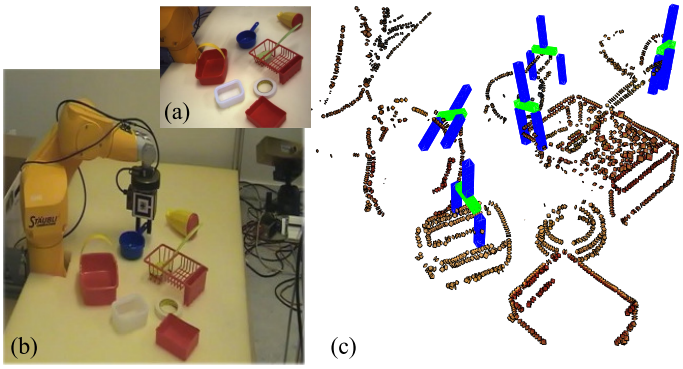

Figure 14: Grasping in a cluttered scene. **(a)** Objects to be grasped. **(b)** The robot grasps an object based on a grasping hypothesis. **(c)** A set of successful grasping hypotheses generated to grasp the objects in the scene.

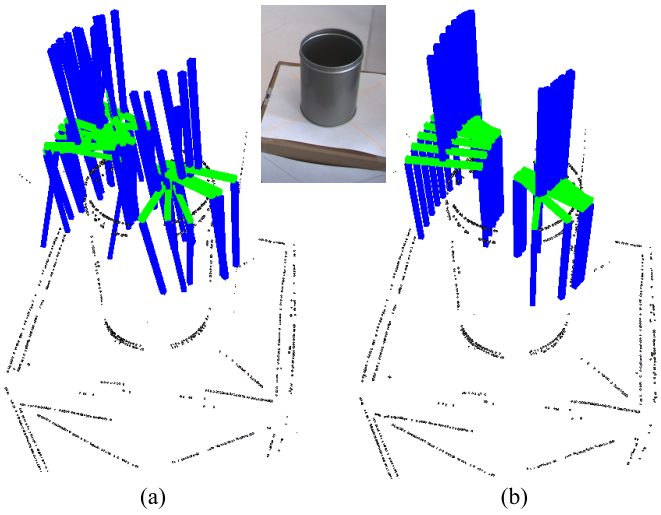

Figure 15: Grasp locations for different approaches. **(a)** Grasps that are calculated by using local features for different noise levels. **(b)** Grasps that are calculated by using contours for different noise levels.

the environment, that is used to guide a mobile robot. In the image domain, the CONDENSATION algorithm uses particle filters to track deformable 2D curves over time [71]. One key aspect of all these methods is that they only encode and track a minimal amount of description (position, sometimes motion), and generally rely on invariance properties from the chosen features to provide robust matching. One important difference of the accumulation method we use is that it transforms and tracks appearance information for each feature in addition to its position in space.

A sample accumulation process is shown in Figure 16(a) where an object is presented to the camera from different perspectives by a robot arm. As shown in Figure 16(b-c), the accumlation process not only creates a full 3D model but also reduces the uncertainty of the 3D primitives. There are two important facts to note in this process. First, the uncertainty of primitives reduces as long as they can be matched in the observation stage. Therefore the uncertainty of contours starts decreasing initially in Figure 16(b) and reaches a stable point where the primitives inside those contours are not updated anymore. Second, we observe that the newly added primitives keep the uncertainty of some contours constant because of their high uncertainty. Note that despite the newly added primitives, the overall uncertainty decreases, as shown in Figure 16(c).

8. Conclusions

We discussed the potential of reasoning with multi-modal contours in the context of scene analysis for driver assistance, depth prediction and robot grasping. In the first application, we have shown the *saliency* and *semantical interpretabilty* of contours and their relations within the context of lane marker detection. In that application, both appearance and geometrical properties as well as inter-relations of contours have been evaluated statistically and found relevant for extracting road lane markers. The application on depth prediction revealed the importance of global reasoning in terms of *reliability* by showing the fact that contours encode richer information than local features in terms of geometry. Similarly, the application on robot grasping pointed to the *reliability* (using contours instead of local features leads to more stable grasping predictions) as well as *saliency* (using coplanar and cocolor contours reduces the

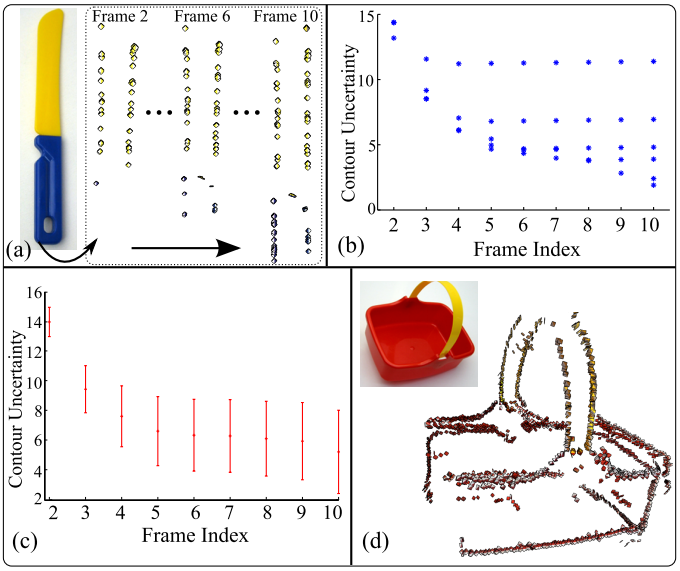

Figure 16: Disambiguation via accumulation. The toy knife is rotated by a robot arm with a known motion, 5 degrees in each frame. **(a)** The accumulation process of a knife. **(b)** Uncertainties of contours during an 8 frames of accumulation. **(c)** Mean and standard variation of uncertainties of contours in each frame. **(d)** An accumulated toy basket.

amount of grasping hypotheses) and *semantical interpretabilty* of contours. In the fourth application, we have shown that the effect of *local uncertainty* and *global uncertainty* problems can be reduced by making use of accumulation over time. Note that the *view-point invariance* property of 3D contour relations plays a crucial role for all four applications since the reasoning takes place in 3D. Also, within the context of driver assistance and grasping we have shown that the 3D contour relations cover a relevant space for learning.

Overall, we observed that the geometrical reasoning in 3D benefits from the use of contours and the uncertainty of the data. Also, we observed that reasoning with global entities decreases computation time of any algorithm making use of relations, and increases the discriminative power of these relations, in particular when reasonable limits for the uncertainty are taken into account.

Acknowledgment

This work was conducted within the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657) funded by the European Commission and the INTERREG 4 A-program Syddanmark-Schleswig-K.E.R.N. with funds by the European Regional Development Fund.

References

[1] O. I. Camps, T. Kanungo, Hierarchical organization of appearance-based parts and relations for object recognition, in: In Proc. IEEE Conf. Computer Vision and Pattern Recognition, IEEE, 1998, pp. 685–691.

[2] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: In IEEE Conference on Computer Vision and Pattern Recognition (CVPR03), pp. 409–415.

[3] A. Sha'asua, S. Ullman, Structural saliency: The detection of globally salient structures using a locally connected network, in: Second International Conference on Computer Vision, pp. 321–327.

[4] J. Shotton, A. Blake, R. Cipolla, Multiscale categorical object recognition using contour fragments, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1270–1281.

[5] D. G. Lowe, Perceptual Organization and Visual Recognition, Kluwer Academic Publishers, 1985.

[6] S. Ullman, High-level Vision, MIT Press, 1996.

[7] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 2 (2004) 91–110.

[8] K. Mykolajczyk, C. Schmid, An affine invariant interest point detector, in: Proceedings of the European Conference in Computer Vision (ECCV 2002), Springer–Verlag, 2002.

[9] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1615–1630.

[10] T. Quack, V. Ferrari, B. Leibe, L. Van Gool, Efficient mining of frequent and distinctive feature configurations, in: ICCV07, pp. 1–8.

[11] A. Mian, M. Bennamoun, R. Owens, A novel representation and feature matching algorithm for automatic pairwise registration of range images, Int. Journal of Computer Vision 66 (2006) 19–40.

[12] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: British Machine Vision Conference, volume 1, pp. 384–393.

[13] P. Moreels, P. Perona, Evaluation of features detectors and descriptors based on 3D objects, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 1, pp. 800–807.

[14] C. Altmann, H. Bülthoff, Z. Kourtzi, Perceptual organization of local elements into global shapes in the human visual cortex, Current Biology 13 (2003) 342–349.

[15] D. J. Field, A. Hayes, R. F. Hess, Contour integration by the human visual system: Evidence for a local association field, Vision Research 33 (1993) 173–193.

[16] M. Behrmann, R. Kimchi, What does visual agnosia tell us about perceptual organization and its relationship to object perception?, Journal of Experimental Psychology: Human Perception and Performance 29 (2003) 19–42.

[17] J. D. Winter, J. Wagemans, Contour-based object identification and segmentation: Stimuli, norms and data, and software tools, Behavior Research Methods, Instruments, and Computers 36 (2004) 604–624.

[18] Y. Lerner, T. Hendler, D. Ben-Bashat, M. Harel, R. Malach, A hierarchical axis of object processing stages in the human visual cortex, Cereb Cortex 11 (2001) 287–297.

[19] K. Tanaka, Mechanisms of visual object recognition: monkey and human studies, Current Opinion in Neurobiology 7 (1997) 523–529.

[20] M. Kikuchi, K. Sakai, Y. Hirai, The mechanism of 3D contour perception, J. Vis. 5 (2005) 76–76.

[21] X. Wang, J. Keller, P. Gader, Using spatial relationships as features in object recognition, Fuzzy Information Processing Society, 1997. NAFIPS '97, 1997 Annual Meeting of the North American (1997) 160–165.

[22] O. Henricsson, Inferring homogeneous regions from rich image attributes, in: Automatic Extraction of Man-Made Objects from Aerial and Space Images, Birkhuser Verlag, 1995, pp. 13–22.

[23] S. Dickinson, A. Pentland, A. Rosenfeld, From volumes to views: an approach to 3-D object recognition, Automated CAD-Based Vision, 1991, Workshop on Directions in (1991) 85–96.

[24] I. Biederman, Recognition-by-components: A theory of human image understanding, Psychological Review 94 (1987) 115–147.

[25] L. G. Shapiro, J. D. Moriarty, R. M. Haralick, P. G. Mulgaonkar, Matching three-dimensional objects using a relational paradigm, Pattern Recognition 17 (1984) 385–405.

[26] S. Fidler, G. Berginc, A. Leonardis, Hierarchical statistical learning of generic parts of object structure, in: CVPR06, pp. 182–189.

[27] S. Fidler, A. Leonardis, Towards scalable representations of object categories: Learning a hierarchy of parts, in: CVPR07.

[28] S. Fidler, M. Boben, A. Leonardis, Similarity-based cross-layered hierarchical representation for object categorization, in: CVPR08, pp. 182–189.

[29] J. Ponce, D. Stam, B. Faverjon, On Computing Two-Finger Force-Closure Grasps of Curved 2D Objects, The International Journal of Robotics Research 12 (1993) 263–273.

[30] J. Speth, A. Morales, P. Sanz, Vision-based grasp planning of 3D objects

by extending 2D contour based algorithms, in: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pp. 2240–2245.
[31] N. Bergström, J. Bohg, D. Kragic, Integration of visual cues for robotic grasping, in: ICVS, pp. 245–254.
[32] N. Krüger, M. Lappe, F. Wörgötter, Biologically Motivated Multi-modal Processing of Visual Primitives, The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour 1 (2004) 417–428.
[33] N. Pugeault, F. Wörgötter, N. Krüger, Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics, International Journal of Humanoid Robotics (Accepted).
[34] J. Clarke, Modelling uncertainty: A primer, Technical Report 2161/98, University of Oxford, Dept. Engineering Science, 1998.
[35] N. Pugeault, S. Kalkan, E. Başeski, F. Wörgötter, N. Krüger, Relations between reconstructed 3D entities, in: Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08).
[36] N. Pugeault, F. Wörgötter, N. Krüger, Multi-modal Scene Reconstruction Using Perceptual Grouping Constraints, in: Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06).
[37] L. Piegl, W. Tiller, The NURBS Book, Springer-Verlag, London, UK, 1995.
[38] I.-K. Lee, Curve reconstruction from unorganized points, Comput. Aided Geom. Des. 17 (2000) 161–177.
[39] J. Hoschek, D. Lasser, Fundamentals of computer aided geometric design, A. K. Peters, Ltd., 1993.
[40] D. R. Murray, Patchlets: a method of interpreting correlation stereo three-dimensional data, Ph.D. thesis, 2004.
[41] D. Murray, J. J. Little, Patchlets: Representing stereo vision data with surface elements, in: WACV-MOTION '05: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1, pp. 192–199.
[42] J. Karhunen, L. Wang, R. Vigario, Nonlinear PCA type approaches for source separation and independent component analysis, in: In Proc. ICNN, pp. 995–1000.
[43] M. Tkalcic, J. Tasic, Colour spaces: perceptual, historical and applicational background, EUROCON 2003. Computer as a Tool. The IEEE Region 8 1 (2003) 304–308 vol.1.
[44] R. Hunt, Measuring Colour. 3rd edition, Fountain Press, Kingston-upon-Thames, 1998.
[45] B. Boesman, L. Jensen, E. Başeski, N. Pugeault, N. Krüger, Bayesian reasoning using 3D relations for lane marker detection, in: Vision, Modeling, and Visualization Workshop 2009 (VMV09).
[46] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, Inc., 1988.
[47] M. Bertozzi, A. Broggi, A. Fascioli, Vision-based intelligent vehicles: state of the art and perspectives., Robotics and Autonomous Systems 32 (2000) 1–16.
[48] J. C. McCall, M. M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, IEEE Transactions on Intelligent Transportation Systems 7 (2006) 20–37.
[49] Y. Wang, E. Teoh, D. Shen, Lane detection and tracking using b-snake, Image and Vision Computing 22 (2004) 269–280.
[50] S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, 3D Lane Detection System based on Stereovision, in: IEEE Intelligent Transportation Systems Conference, pp. 161–166.
[51] S. Kalkan, F. Wörgötter, N. Krüger, Depth prediction at homogeneous image structures, in: VISAPP'08: Int. Conference on Computer Vision Theory and Applications.
[52] S. Kalkan, F. Wörgötter, N. Krüger, First-order and second-order statistical analysis of 3D and 2D structure, Network: Computation in Neural Systems 18 (2007) 129–160.
[53] S. P. Sabatini, G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, K. M. M. V. Hulle, N. Pugeault, N. Krüger, Compact and accurate early vision processing in the harmonic space, International Conference on Computer Vision Theory and Applications (VISAPP) (2007).
[54] J. Ralli, J. Diaz, S. Kalkan, N. Krüger, E. Ros, Disparity disambiguation by fusion of signal- and symbolic-level information, Machine Vision and Aplications (in press).
[55] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, N. Krüger, A strategy for grasping unknown objects based on co-planarity and colour information, Robotics and Autonomous Systems 58 (2010) 551 – 565.
[56] A. Saxena, J. Driemeyer, A. Y. Ng, Robotic grasping of novel objects using vision, Int. J. Rob. Res. 27 (2008) 157–173.
[57] I. Kamon, T. Flash, S. Edelman, Learning to grasp using visual information, in: in Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp. 2470–2476.
[58] K. Huebner, S. Ruthotto, D. Kragic, Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping, in: Proceedings of the 2008 IEEE International Conference on Robotics and Automation, pp. 1628–1633.
[59] L. Bodenhagen, Adaptive Grasping based on Second Order Visual Feature Relations, Master's thesis, University of Southern Denmark, 2009.
[60] N. Pugeault, F. Wörgötter, N. Krüger, Accumulated Visual Representation for Cognitive Vision, In Proceedings of the British Machine Vision Conference (BMVC) (2008).
[61] F. Pilz, N. Pugeault, N. Krüger, Comparison of point and line features and their combination for rigid body motion estimation (2009) 280–304.
[62] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment – A modern synthesis, in: W. Triggs, A. Zisserman, R. Szeliski (Eds.), Vision Algorithms: Theory and Practice, LNCS, Springer Verlag, 2000, pp. 298–375.
[63] P. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, M. Csorba, A solution to the simultaneous localisation and mapping (SLAM) problem, IEEE Transactions in Robotics and Automation 17 (2001) 229–241.
[64] R. van der Merwe, N. de Freitas, A. Doucet, E. Wan, The unscented particle filter, in: Advances in Neural Information Processing Systems (NIPS), volume 13.
[65] J. Guivant, E. Nebot, Optimization of the Simultaneous Localization and Map–Building Algorithm for Real–Time Implementation, IEEE Transactions on Robotics and Automation 17 (2001) 242–257.
[66] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, H. Durrant-Whyte, Simultaneous Localization and Mapping with Sparse Extended Information Filters, International Journal of Robotics Research 23 (2004) 693–716.
[67] T. Lemaire, C. Berger, I.-K. Jung, S. Lacroix, Vision–Based SLAM: Stereo and Monocular Approaches, International Journal of Computer Vision 74 (2007) 343–364.
[68] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem, in: Proceedings of the AAAI National Conference on Artificial Intelligence.
[69] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI).
[70] R. Montúfar-Chaveznava, M. Pérez-Mesa, I. Caldelas, The fly algorithm for robot navigation, in: Research and Education in Robotics – EUROBOT 2008, Communications in Computer and Information Science, Springer Berlin Heidelberg, 2009, pp. 119–127.
[71] M. Isard, A. Blake, Condensation — conditional density propagation for visual tracking, IJCV 29 (1998) 5–28.

Emre Başeski received his B.S. and M.S. degree in Computer Engineering from the Middle East Technical University, Turkey, in 2003 and 2006 respectively. He is currently a Ph.D. student in the Mærsk McKinney Møller Institute, University of Southern Denmark. His research interests include machine vision and learning.

Nicolas Pugeault obtained an M.Sc. from the University of Plymouth in 2002 and an Engineer degree from the Ecole Supérieure d'Informatique, Électronique, Automatique (Paris) in 2004. He obtained his Ph.D. from the University of Göttingen in 2008, and is currently working as a Reseach Fellow at the University of Surrey, United Kingdom.

Sinan Kalkan received his M.Sc. degree in Computer Engineering from Middle East Technical University, Turkey in 2003, and his Ph.D. degree in Informatics from the University of Göttingen, Germany in 2008. He is currently an Assistant Professor in Middle East Technical University. His research interests include biologically motivated Computer Vision, Image Processing and Developmental Robotics.

Leon Bodenhagen received his M.Sc. in Computer Systems Engineering in 2009 from the University of Southern Denmark, where he now is a Ph.D. student at the Mærsk McKinney Møller Institute. His research interests include cognitive systems and cognitive robotics.

Justus H. Piater received the PhD degree from the University of Massachusetts, Amherst, in 2001, where he held a Fulbright graduate student fellowship and, subsequently, spent two years on a European Marie Curie individual fellowship at INRIA Rhône-Alpes, France, before joining the University of Liège, Belgium, where he is a professor of computer science and directs the Computer Vision Research Group. His research interests include computer vision and machine learning, with a focus on visual learning, closed-loop interaction of sensorimotor systems, and video analysis.

Norbert Krüger is a Professor at the Mærsk McKinney Møller Institute, University of Southern Denmark. He holds a M.Sc. from the Ruhr-Universität Bochum, Germany and his Ph.D. from the University of Bielefeld. He is a partner in several EU and national projects: PACO-PLUS, Drivsco, NISA, Handyman. Norbert Krüger is leading the Cognitive Vision Lab which is focussing on computer vision and cognitive systems, in particular the learning of object representations in the context of grasping. He has also been working in the areas of computational neuroscience and machine learning.