# Disambiguating the Recognition of 3D Objects

Gutemberg Guerra-Filho
Department of Computer Science and Engineering
University of Texas at Arlington
416 Yates St., Nedderman Hall,
Arlington, TX 76019-0015
guerra@cse.uta.edu

## Abstract

*We propose novel algorithms for the detection, segmentation, recognition, and pose estimation of three-dimensional objects. Our approach initially infers geometric primitives to describe the set of 3D objects. A hierarchical structure is constructed to organize the objects in terms of shared primitives and relations between different primitives in the same object. This structure is shown to disambiguate the object models and to improve recognition rates. The primitives are obtained through our new Invariant Hough Transform. This algorithm uses geometric invariants to compute relations for subsets of points in a specific object. Each relation is stored in a hash table according to the invariant value. The hash table is used to find potential corresponding points between objects. With point matches, pose estimation is achieved by building a probability distribution of transformations. We evaluate our methods with experiments using synthetic and real 3D objects.*

## 1. Introduction

Object recognition is a fundamental part of artificial vision. This includes surveillance systems, robotics, and several applications such as video annotation, human-machine interaction, virtual reality, object retrieval in databases, image compression, object registration, and others. Although significant advances have been made in this area, several challenges still remain. Addressing these problems involves the development of robust techniques to consider situations where objects are cluttered, occluded, articulated, and changing visual appearance. Furthermore, the observed scene may be sparsely represented with noisy data obtained under different transformations.

To consider these issues, we should take advantage of three-dimensional information and modeling. There is a variety of devices that generate this kind of data by taking a 3D sensory image of the environment. This technology has become more reliable, increasingly widespread, and suitable for real-time applications. Stereo vision, range sensors, time-of-flight cameras, laser scanners, and others are examples of such sensors. Independent of the sensor that observes the environment, a 3D representation of the scene is generated. Consequently, this substantially reduces the need to handle visual appearance changes due to lighting, texture, and viewpoint variation.

In this paper, we address the problems of detection, segmentation, recognition, and pose estimation of 3D rigid objects. Articulated bodies may also be considered assuming each body part consists of a rigid object. Given a cloud of 3D points representing an observed scene and a set of object models, we want to detect any instance of known objects in the scene. The cloud of points is segmented according to the detected objects. Each detected object is recognized as one of the objects in the model set and the corresponding poses are estimated. Our methods do not require any prior segmentation or topological structure such as a mesh. Texture information is not used but could easily be incorporated in our algorithms.

Usually, recognition is performed by comparing the observed scene with a set of object models obtained through a training phase. Among other reasons, recognition might fail due to ambiguity in the set of models, *i.e.*, more than one object model has a strong response to one object in the scene and, possibly, the true object is not the one with the strongest response. In order to overcome this intrinsic ambiguity, our approach initially infers geometric primitives to describe the set of 3D object models. These primitives are learned by finding common geometric patches in different objects. A hierarchical structure is constructed to organize the objects in terms of shared primitives. In this paper, this hierarchy is augmented with relations between different primitives in the same object.

The geometric primitives are obtained through our new Invariant Hough Transform. Given two clouds of 3D points, this algorithm finds the subset of shared points and the transformation required to align these subsets. The algorithm uses geometric invariants to compute relations

978-1-4244-3991-1/09/$25.00 ©2009 IEEE 

for sets of $d$ points (*e.g.*, pairs or triplets). Each relation is stored in a hash table according to an invariant value. A hash table is built for each object in a training set. In a recognition step, the invariant is computed for subsets of $d$ points in the scene. The corresponding sets of points in the hash tables are retrieved and a transformation is computed for each entry retrieved. Each transformation contributes to the computation of a probability distribution for a specific object.

The construction of hierarchical structures to represent a set of objects is usually performed solely by finding common features or patches in different objects. These features are composed into more complex ones in a bottom-up fashion. On the other hand, our approach is a top-down decomposition of the object models, where the main novelty is the use of the relations between points in different primitives of the same object to disambiguate the recognition process. Another innovation proposed in this paper is the Invariant Hough Transform. This is a transform for arbitrary discrete patterns that uses invariant properties to model geometric patterns such as 3D rigid objects and others. Based on this transform, we propose new algorithms to detect, segment, recognize, and estimate the pose of 3D objects.

In summary, the contributions of this work are:

1. A methodology to decompose object models into a set of primitives, organized hierarchically, and to effectively disambiguate the recognition process using invariant intra-relations between points in different primitives of the same object.
2. The Invariant Hough Transform to model 3D objects in terms of invariant properties.
3. Robust and efficient algorithms for detection, segmentation, recognition, and pose estimation of geometric patterns.

In our experiments, we evaluate the robustness and effectiveness of our algorithms using synthetic and real data of 2D and 3D geometric patterns. The experimental results on synthetic data show the good performance of our algorithms with regards to different levels of noise, density of outliers, and data sparseness. We show the effectiveness of our hierarchical structure in disambiguating the geometric patterns by comparing recognition results of tests made with and without our structure. We also discuss the scalability of the augmented hierarchical structure to consider an increasing number of object models. Real 3D data is used to further validate our techniques. These experiments show the efficacy of our algorithms in real cluttered scenes where objects occlude each other. We perform a comparison between our methods and previous work.

The relevance of the work proposed here relies on the novelty of using intra-relations to disambiguate the recognition process. Hough transforms and geometric invariants are two powerful tools in pattern recognition. They are combined here into a single hierarchical comprehensive framework that addresses recognition coupled with detection, segmentation, and pose estimation. This robust framework can be used in cluttered scenes with occluded objects and may be easily generalized to work with composite and articulated objects.

The rest of the paper is organized as follows. In Section 2, we review related work to 3D object recognition. In section 3, we present our Invariant Hough Transform and algorithms for detection, recognition, and pose estimation of geometric patterns. Section 4 describes the hierarchical structure learning process and its application towards disambiguating 3D objects. Experimental results are presented in Section 5. Section 6 discusses our conclusions and research directions.

## 2. Related Work

The construction of a hierarchical structure to organize object models has been used in the context of object recognition through its visual appearance in 2D images. Composite features [3, 17], codebooks [12], segmentation sub-trees [14], and part-based representations [4, 5] are instances of such approaches. In the 3D geometric context, we augment this hierarchical structure with relations between points in different primitives of the same object. This additional structure allows a more effective disambiguation of object models.

Another way to achieve a hierarchical organization for a set of objects uses the Generalized Hough Transform [1, 11, 15]. Variations of the Generalized Hough Transform exploit coherence across consecutive image frames of a video for object tracking [6, 7, 10]. While these methods are able to find common subparts in different 3D objects, they ignore the relations between different parts of the same object. We use invariant properties to account for these relations and explore them towards the disambiguation of the recognition process with our new Invariant Hough Transform. The main differential of our novel transform is that it uses the invariant space as an intermediate step to the parametric pose space.

Other approaches that consider discrete arbitrary patterns and 3D objects are geometric hashing [9, 13, 16] and the Iterative Closest Point algorithm [2, 8]. Geometric hashing matches query objects against a set of object models described by geometric features. The Iterative Closest Point is a technique for the registration of point clouds to a geometric object model that results in its pose estimation. The ICP approach is only used in cases where all cloud points belong to the object. The segmentation of the cloud points into different object models or outliers cannot be performed. A good initial solution for the pose of the 3D object is critical for ICP to converge and the final solution may be a local minimum.

## 3. Invariant Hough Transform

The set $S$ of points in an observed scene is acquired by sampling the surfaces of objects in the scene with a sensor. Each point in $S$ is a noisy measurement of a point on the surface of an object in the scene. Each measured point is non-occluded with respect to the sensor viewpoint. Consequently, only a partial sample of points for each object is obtained from a particular viewpoint. We also assume all measured points are obtained synchronously at the same time.

Given an observed scene, object recognition consists of determining whether any object in a given set $M$ of $m$ object models appears in the observed scene. The recognition process matches object models to the observed scene. The object models are described by a cloud of 3D points in a local coordinate system. Recognition may involve other problems such as detection, segmentation, and pose estimation. While object detection finds objects in the scene just by indicating its coarse localization, segmentation identifies all points in the scene belonging to all detected objects. Pose estimation concerns the computation of the geometric configuration of an object in the scene with regards to its canonical pose in the set of object models. In this paper, we propose algorithms to address the recognition problem coupled with detection, segmentation, and pose estimation.

We combine the use of geometric invariants and Hough transforms, two powerful tools in pattern recognition, into a single framework for 3D object recognition. A geometric invariant is a property whose value is unchanged under a given transformation. For instance, the magnitude of vectors and the area of triangles are invariant under a rigid transformation (rotation and translation). Another example, the interior angles of a triangle and the proportionalities of the lengths of its sides are invariant under rotation, translation, and scaling. Since geometric invariants are independent of an object's current transformation with regards to a canonical pose, invariants allow the comparison of two objects in different poses.

We propose the Invariant Hough Transform (IHT) based on geometric invariants. The IHT allows the recognition of arbitrary non-analytic discrete patterns by accumulating evidence for each object model in a parametric space modeling object pose. Considering rigid transformations in a 3D space, the IHT uses six pose parameters $\{t_x, t_y, t_z, \theta_x, \theta_y, \theta_z\}$, where $(t_x, t_y, t_z)$ is the origin for the 3D shape, and $(\theta_x, \theta_y, \theta_z)$ is its orientation. In a 2D space, the IHT uses three pose parameters $\{t_x, t_y, \theta\}$, where $(t_x, t_y)$ is the origin for the shape, and $\theta$ is its orientation. To find a transformation that aligns two point sets, at least $d$ points are required.

An object recognition approach usually has two phases: an offline acquisition phase where model representations are generated and an online recognition phase where the constructed models are used to recognize the objects in the observed scene.

### 3.1. Offline Acquisition Phase

Given an object model $m_k \in M$ consisting of $n_k$ points $\{p_1, p_2, \ldots, p_n\}$, a tuple $(p_{i_1}, \ldots, p_{i_d})$ is a list of $d$ points in $m_k$, where $1 \le k \le m$ and $1 \le i_1, \ldots, i_d \le n_k$. Each different tuple is a combination of $d$ points in the object model. A tuple of points is associated with an invariant value $\phi(p_{i_1}, \ldots, p_{i_d})$. For example, considering rigid transformations, an invariant for a pair of points is the Euclidean distance between the two points.

For each possible tuple in the object model with an invariant value $\phi(p_{i_1}, \ldots, p_{i_d})$, its quantized invariant value $v$ serves as an index to a bin in a hash table $H$ where an entry associated with the model $m_k$ and tuple $(p_{i_1}, \ldots, p_{i_d})$ is stored at $H(v)$. Each hash table bin contains a list of entries of the form $(m_k, p_{i_1}, \ldots, p_{i_d})$. A pre-processing step encodes each model $m_k$ in the set $M$ as a hash table $H$ obtained for all possible tuples $(p_{i_1}, \ldots, p_{i_d})$ in $m_k$. The time complexity of the pre-processing step is $O(mn^d)$, where $m$ is the number of object models, $n$ is the maximum number of points in a model, and $d$ is the number of points needed to form a tuple.

### 3.2. Online Recognition Phase

In the recognition phase, an observed scene consists of a cloud of points $S$. Each tuple $(q_1, \ldots, q_d)$ of points in the set $S$ corresponds to an invariant value $\phi(q_1, \ldots, q_d)$ as described above. Its quantized invariant value corresponds to hash table index $v$. Each entry $(m_k, p_{i_1}, \ldots, p_{i_d})$ in the bin $H(v)$ is a possible match to the tuple $(q_1, \ldots, q_d)$ of points in the scene. This way, invariant values are used to find possible correspondences between points in the scene and points in the object models, such that $q_j \in S$ corresponds to $p_{i_j} \in m_k$, where $1 \le j \le d$.

From this correspondence, we compute a transformation $T$ that maps each measured point $q_j$ to point $p_{i_j}$. Basically, this transformation ideally aligns the object pose in the scene with its canonical pose in the model set. This transformation considers a single possible correspondence between a scene tuple and a hash entry tuple. The corresponding points are used to solve for the transformation parameters by solving a set of equations of the form $q_j = Tp_{i_j}$.

The solution transformation $T$ gets a vote by incrementing the corresponding bin in an accumulator array $A_k$. An accumulator is a discrete storage count associated with the parametric space. This way, the accumulator $A_k$ represents a probability distribution for a transformation concerning the mapping of object model $m_k$

into its possible instances in the scene. Note that an accumulator is a six-dimensional array to account for transformations in a 3D space or a three-dimensional array when considering transformations in a 2D space. Each accumulator $A_k$ is also associated with a list array $L_k$ where each entry in $L_k$ is a list of scene point tuples that contributed with votes for the corresponding entry in $A_k$. Note that the entry values in $A_k$ are simply the size of the corresponding lists in $L_k$.

The procedure above is repeated until enough transformation solutions are counted so a meaningful distribution in parametric space is supported (see Fig. 1). The maximum count in the accumulator $A_k$ is associated with the most relevant transformation and the entry coordinates correspond to the pattern parameters, an absolute pose for the instance of object model $m_k$ in $M$.

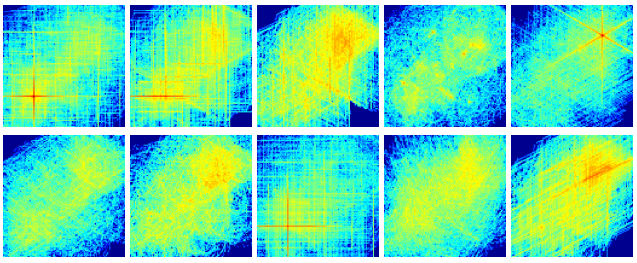

Figure 1. Two-dimensional accumulators for an IHT considering only translation $\{t_x, t_y\}$. The scene contains two patterns indicated by peaks in the top left and top right arrays.

If we consider all possible tuples of points in the scene, the time complexity of the recognition phase is $O(hs^d)$, where $h$ is a hash table bin processing time and $s$ is the number of points in the observed scene. However, instead of checking for all $O(s^d)$ possible tuples in the scene, we use a more efficient randomized iteration where only a significantly smaller number of random tuples are needed. Furthermore, parallelization can be achieved by using a different hash table for each object model. This way, the online recognition algorithm is very efficient.

### 3.3. Detection, Segmentation, and Pose Estimation

Once the recognition process is performed; object detection, segmentation, and pose estimation is obtained from the accumulators and associated list arrays.

In a 3D space, for example, *object detection* is achieved by finding the entry $A_k(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$ with the maximum value in all accumulators $A_k$ for $1 \le k \le m$. This way, the object $m_k$ is detected in the observed scene. *Object segmentation* identifies all points in the scene belonging to the detected object $m_k$. The set of scene points belonging to this instance of $m_k$ is retrieved as the set of points in all tuples in the list $L_k(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$. Note that later we may detect additional instances of $m_k$ if another maximum is found in $A_k$. The pose estimation amounts to use $\{t_x, t_y, t_z, \theta_x, \theta_y, \theta_z\}$ as a coarse estimation and to refine this initial solution with a least squares or ICP method.

Once object $m_k$ is detected in the scene, we update the accumulators and list arrays to remove all scene points associated with this instance of $m_k$. The detection, segmentation, pose estimation process is repeated until all points in the scene are segmented or the maximum value in all accumulators is less than a threshold value.

## 4. Augmented Hierarchical Structure

In general, recognition is achieved by considering each model acquired in a pre-processing training step and selecting the model that best fits the observed object. However, due to the complex nature of the problem and to an intrinsic ambiguity among patterns, the strongest response among all patterns may not be the true answer.

Assume $m_k$ is the object model corresponding to the actual object in an observed scene. We measure ambiguity as the ratio of the strongest response $\alpha$ of any model but the true model $m_k$, $\alpha = \max(A_l)$ for $l = 1, \ldots, m$ and $l \neq k$, to the response $\alpha_k$ of the true object model, $\alpha_k = \max(A_k)$. For a particular uncertainty value $\varepsilon > 0$, if the ambiguity ratio $\alpha/\alpha_k$ is lower than $1\text{-}\varepsilon$, the recognition process results in the true object. Otherwise, the recognition process is compromised. Due to ambiguity and uncertainty, the outcome may be incorrect and the process may result in the wrong object being recognized. Uncertainty happens due to several reasons such as noise, outliers, sparse input, occlusions, a cluttered scene, and deformations. These issues are handled by developing robust algorithms, but we also have to reduce ambiguity in order to improve the effectiveness of recognition methods.

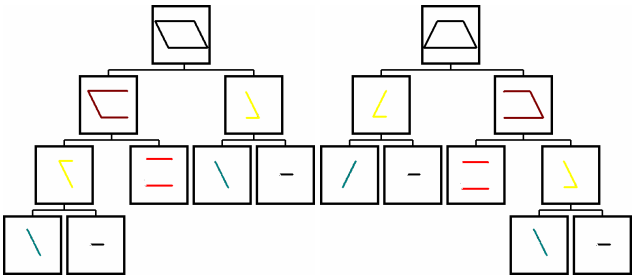

Figure 2. Two objects in the set of object models represented as a learned hierarchical structure in terms of geometric primitives.

To reduce the intrinsic ambiguity among object models, we construct a hierarchical structure that organizes the object models in terms of primitives. Figure 2 shows part of an actual hierarchy learned by our algorithms. The structure is based on common primitives shared by

different objects. This way, this hierarchical structure models the inter-relations between different objects. However, just finding common primitives in different object patterns and its inter-relations may not be enough to disambiguate the recognition process. Therefore, we augment the hierarchical structure with relations between points in different primitives of the same object. These intra-relations are obtained as invariant values for tuples of points in the same object but in different primitives.

### 4.1. Augmented Hierarchy Construction

The construction of our augmented hierarchical structure uses a hash table $H_k$ for each object model $m_k$ in $M$, accumulators $A_{a,b}$ and the associated list arrays $L_{a,b}$ for all pairs of object models $m_a$ and $m_b$, where $1 \le a < b \le m$. The initialization of these data structures is performed in a manner similar to the Invariant Hough Transform algorithms.

**Initialization.** The offline acquisition step of the IHT is adapted to find the hash tables $H_k$ for each object model $m_k$ in $M$. For each object model $m_k \in M$ and for each possible tuple $(p_{i1},\ldots,p_{id})$ of points in $m_k$, the quantized value $v$ of an invariant $\phi(p_{i1},\ldots,p_{id})$ serves as an index to a bin in the hash table $H_k$ where an entry associated with the tuple $(p_{i1},\ldots,p_{id})$ is stored at $H(v)$. The time complexity of this pre-processing step is $O(mn^d)$.

The online recognition step of the IHT is adapted to compute the accumulators $A_{a,b}$ and the associated list arrays $L_{a,b}$ for each pair of object models $m_a$ and $m_b$ in $M$. Each tuple $(q_1,\ldots,q_d)$ of points in $m_a$ corresponds to an invariant value $\phi(q_1,\ldots,q_d)$. Its quantized invariant value corresponds to hash table index $v$. Each entry $(p_{i1},\ldots,p_{id})$ in the bin $H_b(v)$ is a possible match to the tuple $(q_1,\ldots,q_d)$, such that $q_j \in m_a$ corresponds to $p_{ij} \in m_b$, where $1 \le j \le d$. From these correspondences, we compute a transformation $T$ that maps each point $q_j$ to point $p_{ij}$. The solution transformation $T$ gets a vote by incrementing the corresponding bin in an accumulator array $A_{a,b}$. Each accumulator $A_{a,b}$ is also associated with a list array $L_{a,b}$ where each entry in $L_{a,b}$ is a list of pairs of tuples with points in $m_a$ and $m_b$, respectively, that contributed with votes for the corresponding entry in $A_{a,b}$. A tuple pair has the form $[(q_1,\ldots,q_d),(p_{i1},\ldots,p_{id})]$. The time complexity of this pre-processing phase is $O(hm^2n^d)$.

**Structure finding.** In an iterative step, we find the geometric primitive with the strongest response in all accumulators. A new hash table and the other associated data structures are created to represent the discovered primitive as a new model. The old hash tables, accumulators, and list arrays are also updated to represent the removal of the points in the new primitive from the two models sharing this primitive.

In each iterative step, first we find the entry $A_{a',b'}(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$ with the maximum count in all accumulators $A_{a,b}$ for $1 \le a < b \le m$. A new geometric primitive $m_\delta$ shared by model $m_{a'}$ and $m_{b'}$ is associated with this entry (see Fig. 3). The set of points in $m_{a'}$ (and in $m_{b'}$) belonging to the new object primitive $m_\delta$ is retrieved as the set of points in the tuple with points in $m_{a'}$ of all tuple pairs in the list $L_{a',b'}(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$. In the augmented hierarchy, we record that $m_\delta$ is a part of objects $m_{a'}$ and $m_{b'}$. The primitive $m_\delta$ and its instances in the original objects $m_{a'}$ and $m_{b'}$ are related by the identity transformation and by the transformation built with parameters $(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$, respectively.

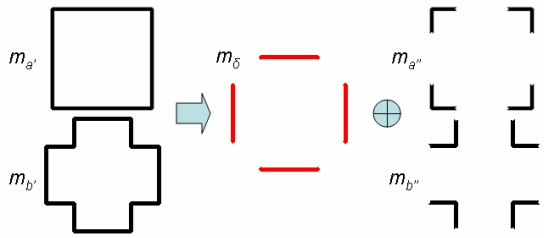


Figure 3. A new geometric primitive $m_\delta$ is discovered in object models $m_{a'}$ and $m_{b'}$.

Once the new object $m_\delta$ is created, we create a new hash table $H_\delta$, the accumulators $A_{a,\delta}$ and respective list arrays $L_{a,\delta}$ for $1 \le a \le m$. We also update the existing hash tables $H_{a'}$ and $H_{b'}$; the accumulators $A_{a',b}$ for $1 \le a' < b \le m$ ($A_{a,a'}$ for $1 \le a < a' \le m$) and $A_{b',b}$ for $1 \le b' < b \le m$ ($A_{a,b'}$ for $1 \le a < b' \le m$); and the associated list arrays. This update is necessary to remove all points associated with the new model $m_\delta$ from the data structures related to models $m_{a'}$ and $m_{b'}$.

Let the set $U_{a'}$ be the set of all tuples in all entries of the hash table $H_{a'}$. The set $U_{a'}$ can be divided into three sets of tuples $U^+_{a'}$, $U^-_{a'}$, and $U^0_{a'}$; where $U^+_{a'}$ contains tuples with all points in $m_\delta$, $U^-_{a'}$ contains tuples with all points not belonging to $m_\delta$, and $U^0_{a'}$ contains tuples with some points in $m_\delta$ and other points not in $m_\delta$ (see Fig. 4). The new hash table $H_\delta$ is constructed from $U^+_{a'}$ and the updated hash table $H_{a'}$ is obtained from $U^-_{a'}$. The hash table $H_{b'}$ is updated in a similar way. The set $U^0_{a'}$ represents the intra-relations between points in different primitives of $m_{a'}$. This is an additional structure kept as a hash table $H_{a',\delta}$ to disambiguate the online recognition process. The object models are organized in terms of the geometric primitives and the internal relations are used to discern between similar but different patterns.

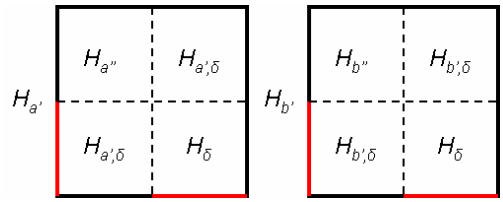


Figure 4. Consider a hash table with 2D tuples ($p_{ix}$, $p_{iy}$) organized into axis representing a point ($ix$, $iy$) in the tuple. The 2D entries in the hash tables $H_{a'}$ and $H_{b'}$ are divided considering whether points are in the geometric primitive $m_\delta$ (red) or not (black).

The creation and updates of accumulators and list arrays are performed analogously. We consider a list array $L_{a',b}$ as the set $U_{a',b}$ of tuple pairs in the lists of all entries in the array. Again, this set of tuple pairs is partitioned into three sets $U^{+}_{a',b}$, $U^{-}_{a',b}$, and $U^{0}_{a',b}$ according to a *base tuple* with points in $m_{a'}$ (the other tuple in each pair contains points in $m_b$); where $U^{+}_{a',b}$ contains tuple pairs with all base tuple points in $m_\delta$, $U^{-}_{a',b}$ contains tuple pairs with all base tuple points not belonging to $m_\delta$, and $U^{0}_{a',b}$ contains tuple pairs with some base tuple points in $m_\delta$ and other base tuple points not in $m_\delta$. The new list array $L_{b,\delta}$ is constructed from $U^{+}_{a',b}$ and the updated list array $L_{a',b}$ is obtained from $U^{-}_{a',b}$. Accumulators only reflect the size of these newly created and updated list arrays.

The discovery of new primitives and incremental construction of the augmented hierarchical structure is repeated until the maximum value in all accumulators is less than a threshold value.

### 4.2. Disambiguated Online Recognition

The augmented hierarchical structure is applied to reduce the ambiguity in the online recognition steps. The disambiguated online recognition process uses the hash tables $H_k$ for the geometric primitives and the intra-relations $H_{k,\delta}$ constructed as described above. The hierarchy of primitives is used to aggregate evidence from the lowest level to the highest level.

Initially, we compute the accumulators $A_k$, $A_{k,\delta}$ and the associated list arrays $L_k$, $L_{k,\delta}$ using the corresponding hash tables for geometric primitives $H_k$ and intra-relations $H_{k,\delta}$. According to these hash tables, each tuple $(q_1,\ldots,q_d)$ of points in the observed scene $S$ corresponds to votes in the accumulator arrays.

The hierarchical decomposition of patterns into primitives iteratively divides the patterns $m_{a'}$ and $m_{b'}$ sharing the same primitive $m_\delta$ into three parts: $m_{a''} = m_{a'} - m_\delta$, $m_{b''} = m_{b'} - m_\delta$, and $m_\delta$. In a bottom-up direction, we compute the accumulator $A_{a'}$ from the accumulators associated with primitive $m_{a''}$ and intra-relation $H_{a'',\delta}$: $A_{a'} = A_{a''} + 2{*}A_{a'',\delta}$. Similarly, the accumulator $A_{b'}$ is computed from the accumulators associated with primitive $m_{b''}$ and intra-relation $H_{b'',\delta}$: $A_{b'} = A_{b''} + 2{*}A_{b'',\delta}$. Note that the accumulator $A_\delta$ associated with hash table $H_\delta$ is eliminated from the evidence accumulation to reduce ambiguity (see Fig. 5).

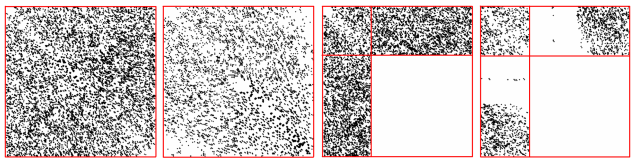

Figure 5. The hash table accumulation for two object models according to plain IHT (left) and augmented hierarchical (right).

The aggregation process is repeated for all higher level primitives. This way, accumulators and respective list arrays are built for all primitives in the hierarchy. Once the highest level data structures are found; recognition, detection, segmentation, and pose estimation are performed in a similar way as described for the IHT.

## 5. Experimental Results

In our experiments, we evaluate the robustness and effectiveness of our algorithms using synthetic and real data of 2D and 3D geometric objects.

### 5.1. Two-Dimensional Synthetic Data

We generated 2D synthetic data based on a set of geometrical shapes (see Fig. 6). We selected 2D shapes as a motif to obtain a set of 2D objects with extreme ambiguity. In our experiments, the synthetic data allows the control of the noise level, the number of outliers, and the density of the data. The 2D objects are retrieved from images. The cloud of points describing the object models are the border points without connectivity or topology. We considered 2D objects mostly consisting of regular polygons and closed curves. Our agenda was to construct a set of objects with a considerable amount of ambiguity. This way, we used this 2D synthetic data to evaluate our algorithms with regards to noise, outliers, and partial information.

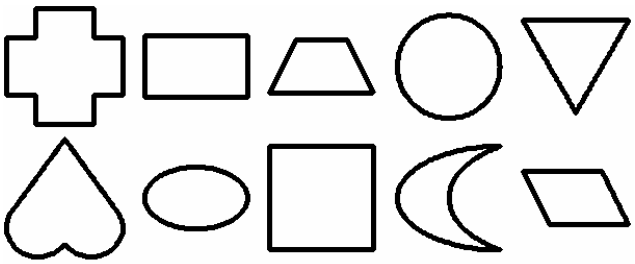

Figure 6. Two-dimensional synthetic data.

**Robustness against Noise, Outliers, and Sparse Data.** To evaluate our methods with respect to the level of noise, we perturb the point coordinates of synthetic scenes. We performed online recognition experiments where noise is modeled as a normal distribution with mean zero and standard deviation one. The noise is amplified by a constant that ranges from 0 (no noise) to a percentage of the diameter of the object. The diameter is the distance between the two furthest points in the object model. Outliers are inserted as spurious points according to a uniform distribution. The quantity of outliers ranges from 0 to 2.0 times the number of points in the object. The density also varies from 0 to 100% of the original points.

Figure 7 shows the results of our experiments in terms of ambiguity ratio. A ratio less than 1-$\varepsilon$ means a correct recognition. Our algorithms were reliable with noise amplified by up to 7% of the radius of the object considered. The algorithm is not affected by outliers and demands a small density to perform satisfactorily.

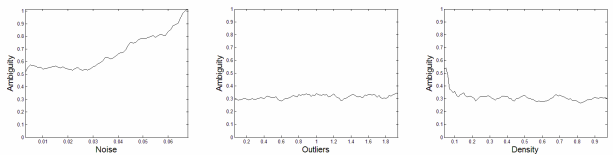


Figure 7. Robustness evaluation.

**Disambiguation and Scalability.** The effectiveness of our augmented hierarchical structure in disambiguating the geometric patterns is shown by comparing recognition results of tests made with and without our structure. We performed several recognition tests and noticed a significant reduction in the ambiguity ratios. Figure 8 shows some recognition tests where, for example, a single hierarchical decomposition reduced the ambiguity in the plain IHT (blue) from 39.47% to 8.90% with the augmented hierarchical structure (red).

The augmented hierarchical structure may consider an increasing number of object models without affecting its performance. The addition of a new object only affects the recognition of similar objects. The scalability is due to the independence of each object model in the recognition process disambiguated by primitives and intra-relations.

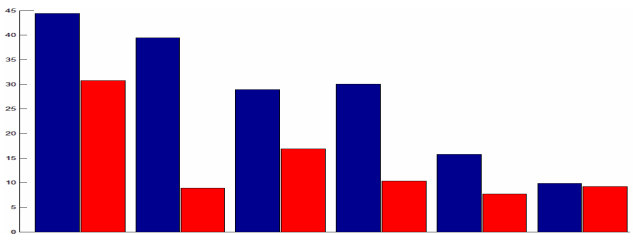

Figure 8. Reduced ambiguity.

## 5.2. Three-Dimensional Real Data

Real 3D data are used to further validate our methods. These experiments show the efficacy of our algorithms in real cluttered scenes where objects occlude each other.

We collected three-dimensional data from eight objects using a 3D laser scanner. Our collection of 3D objects consists of a basketball, a billiard chalk, a medicine cup, a dental floss, a memory stick, a pain reliever box, a spice container, and a stone (see Fig. 9). The largest object was the pain reliever box with dimensions 9cm by 4.5cm by 45mm. The captured object model contains a cloud of 8,752 points which corresponds to a density of 54 points per $cm^2$ (349 points per squared inch).

Figure 9. Three-dimensional objects.

**Occlusion and Clutter.** We evaluated our algorithms with realistic situations where objects are cluttered and occluded. A cluttered scene contains several objects with possibly more than one instance of the same object. The objects in the scene may partially occlude each other. We experimented with four complex scenes capture with a single view scan (see Fig. 10).

Figure 10. Cluttered scenes with occluded objects.

Each scene contains three objects in our collection and, in the first scene, an additional unknown object. The first scene captures a memory stick, a dental floss, a billiard chalk, and an unknown bottle cap with 606 points. The second scene captures a pain reliever box, a billiard chalk, and a basketball with 836 points. The third scene observes

a stone, a medicine cup, and a spice container with 554 points. The fourth scene contains a medicine cup, a basket ball, and a dental floss with 814 points.

In the three-dimensional space, we consider rigid transformations. The correspondence between at least three 3D points in different sets of points is required to compute the rigid transformation that aligns these clouds of points representing a rigid object. Consequently, our 3D implementation of the augmented hierarchy for the IHT uses tuples of three 3D points. The invariant used was the Euclidean distance between the points in the tuple.

Since pose estimation accuracy was mainly dependent on the size of the accumulators, results are shown as the cloud segmentation for each scene (see Fig. 10). Each object model in the highest level and geometric primitive in the lowest level function as an independent process of recognition. These parallel recognition processes performed well in the realistic cluttered scenes and the response was adequate for the amount of occlusion.

**Comparison.** As a final experiment, we compare our Invariant Hough Transform (plain and hierarchical) to Geometric Hashing. A set of 20 objects is used to obtain ambiguity ratios (see Fig. 11). Our comparison shows that IHT achieves better results than Geometric Hashing and hierarchical IHT is effective in further reducing ambiguity.

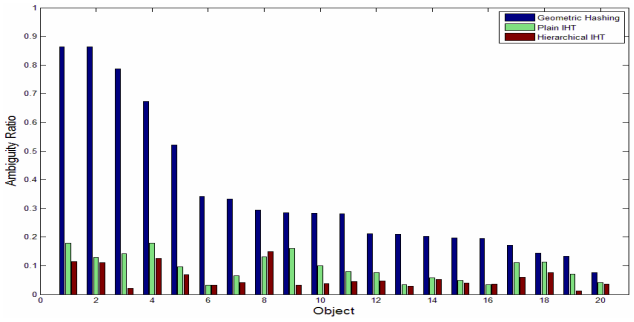


Figure 11. Comparison of Geometric Hashing and IHT.

## 6. Conclusion

The Invariant Hough Transform is a powerful concept that generalizes to other invariants, including projective and photometric invariants. This way, we may apply the same methods to recognition based on shape and appearance in 2D, 3D, projective, and photometric spaces.

As future work, we want to generalize our framework to consider deformation and categorization. We also foresee applications to composite and articulated objects such as the human body. This way, the augmented hierarchy could be used in human pose estimation and, consequently, action recognition.

In this paper, we proposed a novel Invariant Hough Transform for the recognition of 3D objects. We use a hierarchical structure that is augmented with intra-relations. The framework is able to reduce ambiguity and recognize objects in cluttered scenes with occluded objects under noise, outliers, and a low density of points.

The relevance of the proposed work relies on the originality of using intra-relations to disambiguate the recognition process. The advance of recognition algorithms usually takes place in two fronts: the development of robust methods to handle uncertainty and the organization of sensory data to avoid ambiguity. We believe this work provides contributions in both directions.

## References


[1] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2): 111-122, 1981.

[2] P. Besl and N. McKay. A method for registration of 3-D shapes. *PAMI*, 14(2): 239-256, 1992.

[3] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In Proc. of ICCV, vol. 1, pp. 220-227, 2005.

[4] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1): 55-79, 2004.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In Proc. of CVPR, vol. 2, pp. 264-271, 2003.

[6] M. Greenspan, L. Shang, and P. Jasiobedzki. Efficient tracking with the bounded Hough transform. In Proc. of CVPR, vol. 1, pp. 520-527, 2004.

[7] L. Iocchi, D. Mastrantuono, and D. Nardi. A probabilistic approach to Hough localization. In Proc. of ICRA, vol. 4, pp. 4250-4255, 2001.

[8] S. Knoop, S. Vacek, and R. Dillmann. Modeling joint constraints for an articulated 3D human body model with artificial correspondences in ICP. In Proc. of Humanoids, pp. 74-79, 2005.

[9] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In Proc. of ICCV, pp. 238-249, 1988.

[10] P. Lappas, J. Carter, and R. Damper. Object tracking via the dynamic velocity Hough transform. In Proc. of ICIP, vol. 2, pp. 371-374, 2001.

[11] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In Proc. of BMVC, pp. 759-768, 2003.

[12] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In Proc. of ECCV, pp. 575-588, 2006.

[13] J. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *IJRR*, 6(2): 29-44, 1987.

[14] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In Proc. of CVPR, vol. 1, pp. 927-934, 2006.

[15] M. Ulrich, C. Steger, and A. Baumgartner. Real-time object recognition using a modified generalized Hough transform. *Pattern Recognition*, 36(11): pp. 2557-2570, 2003.

[16] H. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *Computational Science & Engineering*, 4(4): 10-21, 1997.

[17] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In Proc. of CVPR, vol. 2, pp. 1491-1498, 2006.