

# Recognition using Specular Highlights

Aaron Netz and Margarita Osadchy, *Member, IEEE*,

## Abstract

We present a novel approach to pose estimation and model-based recognition of specular objects in difficult viewing conditions, such as low illumination, cluttered background, large highlights and shadows that appear on the object of interest. In such challenging conditions conventional features are unreliable. We show that under the assumption of a dominant light source, specular highlights produced by a known object can be used to establish correspondence between its image and the 3D model, and to verify the hypothesized pose and the identity of the object. Previous methods that use highlights for recognition make limiting assumptions such as known pose, scene-dependent calibration, simple shape, etc. The proposed method can efficiently recognize free form specular objects in arbitrary pose and under unknown lighting direction. It uses only a single image of the object as its input and outputs object identity and the full pose. We have performed extensive experiments for both recognition and pose estimation accuracy on synthetic images and on real indoor and outdoor images.

## Index Terms

Object recognition, varying illumination, pose estimation, invariants, specularities.

## I. INTRODUCTION

We present a model-based method for recognition of specular objects under arbitrary pose and lighting direction. Given a database of predefined 3D object models, and a single 2D image of one of the objects in the database, the task is to determine which object appears in the image. Much work has been done on model-based recognition of Lambertian objects with prominent texture or shape features (e.g, [28, 29, 22, 21, 38, 4, 9, 12, 11]). The Lambertian assumption helps in dealing with illumination effects, while prominent texture and shape features allow to find good correspondences or perform indexing. Very little of these is available in images

A. Netz and M. Osadchy are with the Computer Science Department, University of Haifa

of smooth, glossy, textureless objects with highlights and shadows, which are placed against a cluttered scene. Figure 5 shows examples of such images. Glossy smooth objects produce specular effects and unlike texture, specular highlights do not remain constant under changes of viewpoint. Although not constant, the specular highlights obey, under certain assumptions, some rules of consistency, which makes them useful for recognition. Specular highlights have several advantages over conventional features: they are easy to detect even by simply thresholding the image and they are robust to changes in background, texture variation, and occlusion of non-highlighted parts. In addition, they can be used with transparent objects, where extracting contours or similar features is very hard. Thus rather than filtering out specular highlights as was done in most previous methods, we use the highlights produced by a known object, to extract information that can assist in recognition. Specifically, we show that highlights produced by an object in images that differ by lighting and viewing directions, are related by an approximately affine transformation (Figure 1). We exploit this observation to establish correspondences between an object’s image and the 3D model. We then use the correspondences to compute the pose of the object and verify the pose and the identity by measuring the similarity between the specular highlights extracted from the input image and the highlights predicted for the hypothesized pose and identity, using a simple model of highlight formation proposed in [34].

Since many real objects have highlights, a practical recognition system should be robust to these effects. Our method could be used in applications where objects are very specular and conventional features are unreliable. When such features are available, they can be incorporated as additional correspondences into the proposed framework. Examples of applications are monitoring of manufacturing, defect detection, or domestic applications, such as vision systems for assisted living. 3D models can be acquired using stereo or structured light systems, if shiny objects are first covered with powder to reduce their shininess. In industrial applications (manufacturing and inspection), CAD models of the objects are available. In domestic applications, some objects such as kitchenware have a standard shape and are specular.

The main contributions of this work are:

- We show that highlights produced by a surface patch in images that differ by lighting and viewing directions, are related by an approximately affine transformation.
- We use highlights to establish correspondences for pose estimation which is much more efficient than a brute force search done by previous methods for pose estimation of specular

objects (e.g.,[10]).

- We propose a method for recognition of specular objects under arbitrary pose and lighting direction that requires only a single image and doesn't require any calibration object or procedure for estimating illumination in the scene.
- Our method can work with shapes that are much more complex than those used in previous works ([10, 2]).

We make the following assumptions throughout this paper:

- 1) The scene is lit by a single, distant light source of unknown directions.
- 2) The input contains a single object to be recognized, of unknown position and orientation in 3D-space.
- 3) The object of interest is relatively distant from the camera.
- 4) The object of interest has an approximately uniform reflectance.
- 5) At least 3 significant and distinct specular highlights are visible in the image.

The last assumption is relevant if the recognition is done using solely specular features. If an image contains other more conventional features (contours, lines, etc.), but only one or two highlights, our approach can easily incorporate correspondences obtained from the conventional features for pose computation, while the verification phase remains unchanged.

The assumption of dominant light source holds for outdoor scenes. It doesn't hold for indoor scenes, which are illuminated by many sources or extended lights. However, in such conditions the object is well illuminated and even if it has highlights, existing methods (e.g., contour based methods [4, 23]) could work fairly well. Images taken with a directional light are poorly illuminated and have large shadows, which makes recognition much more difficult.

Assumptions (2) and (3) usually hold or can be resolved by adding a preprocessing step to a recognition system (e.g., a module for extracting a specular object from its background [13]). Currently we do not explicitly model interreflections, cast shadows, and occlusions. However when these effects are minor, we treat them as noise. We test our algorithm on real objects, achieving good results even when the assumptions do not precisely hold.

The experiments presented in this paper are performed on synthetic and real objects. We constructed a database of real, complex objects which includes CAD models and images of these objects under variation of pose, background, and illumination direction (including indoor and outdoor illumination). This data set is much more diverse than those used in previous papers.

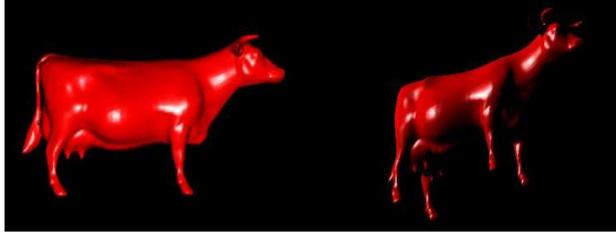


Fig. 1. Left: offline-rendered frontal view of the object with frontal light, right: a given picture of the object in an unknown view with unknown light direction. In both the highlights are produced by the same surface patches and thus highlights undergo affine transformation and could be used for establishing correspondence.

The experiments (see Section V) show good performance considering that our method uses very little information about the scene and only a few percents of the input image – the highlights.

The paper is organized as follows. Section II discusses previous work on specularities and provides an overview of the methods that are used in our approach. Section III describes the basic approach, including its theoretical background. Section IV shows an example of a system for recognition of specular objects. Section V describes the new database of real and synthetic images of specular objects and the experiments performed on this data base. Section VI concludes the paper and discusses future directions.

## II. RELATED WORK

In this section we first review the previous work on using specular cues for pose estimation and object recognition and then we discuss several methods for specularities extraction and pose estimation using correspondences, which can be integrated with the proposed approach.

### A. Using Specular Cues

There has been much work aimed at analyzing specular effects in images, most of which has been concerned with surface reconstruction [6, 7, 43, 32, 49, 41, 37, 47]. Recently, several methods have been proposed for detecting specular surfaces in images [30, 13].

Specular highlights reveal accurate local information about the shape of the object. Thus a natural idea is to use them for alignment. This idea was employed in [27], which showed very impressive results.

Only very recently specular highlights have been used in pose estimation [2, 10]. The method presented in [2] incorporates different channels of information, one of which is a polarization angle of the light reflected from the object surface that provides information on the rotation of

an object relative to the camera. The data acquisition process for this method is quite involved. It includes taking many images with different shutter times to create a high dynamic range image, two images for depth estimation, one with small aperture and another with large, and it also needs a polarizer. Finally, all parts in this method require calibration. The method presented in [10], addresses 3D pose estimation and segmentation of specular object. It proceeds by rendering images of highlights for every viewing direction using the environmental mapping acquired by placing a mirror ball in the scene. These images are used in a brute-force search for 5 pose parameters (the distance to the camera is assumed known), producing a rendering that most resembles the input image. The pose is found by first searching for the best translation for each orientation using a standard optimization with an energy function based solely on highlights. The translation is refined by removing the pixels with low elevation of incident light (to reduce the effect of interreflections). The rotation with minimal cost is chosen and then all 5 parameters are refined by maximizing the correlation of the input and rendered intensity images (excluding pixels with low elevation of light). The experiments presented in [10] are done on simple objects with complex illumination, which strongly constrains the appearance of highlights. The same work [10] proposes to use specular flow instead of an environment map but still using a brute-force search. In order to compute the specular flow they require angular motion of far-field environment, which is also a limiting requirement.

Norman et. al. [31] showed empirically that specular highlights provide a significant aid in human perception of 3D shape. Nevertheless, due to the difficulty of the task, very little work has been done on recognition of specular objects [40, 18, 34]. In pioneering work on the recognition of shiny objects [5, 15] the main theme has been to look for specularities with specific shapes that are likely to appear on objects of interest. For example, cylinders were detected using elongated specularities [5]. Another direction has been to use specularities to infer the 3D shape of the corresponding regions and then match these region to a known 3D model. Such an approach was taken by Koshikawa and Shirai [25] in their recognition system that uses polarized illumination to estimate the surface normals of an object. Georghiades [16] used the Torrance-Sparrow model to recover the shape and the non-Lambertian reflectance properties of faces. These were used to model the appearance of each face under variable lighting and viewpoint. Sato et al.[39] employed a physics-based simulator to predict specularities from which a set of aspects of the object was generated. For each specular aspect they constructed deformable matching templates. At runtime,

an input image was first classified into candidate aspects, then the deformable templates were used to refine matching. The method proposed in [18] also operates on specular features and uses multiple observations from different viewpoints for resolving an ambiguity in the recognition of specular objects.

Model-based methods for recognition of Lambertian objects account for illumination effects by building low dimensional representations of the set of images that a 3D object produces under a large range of lighting conditions. Basri and Jacobs [3] and Ramamoorthi and Hanrahan [36] showed how to analytically derive a 9D representation of an object’s image set from a 3D model. Light reflected by specular objects has a higher spatial frequency [48], implying that modelling their appearance requires many more harmonics. Using more harmonics results in heavy computations, noisy solutions, or even significant errors. Shirdhonkar and Jacobs [44] approached these problems by enforcing the nonnegativity constraint of light in lighting recovery. The method is based on the extension of Szegos eigenvalue distribution theorem to spherical harmonics and is formulated as SDP. Osadchy et al [34] propose a very different approach for recognition of specular objects in which they exploit Lambertian reflection and highlights as separate cues. They used specular highlights to determine whether an image fits a given model of a known pose, based on the observation that the normal vectors of specular points should map to a small disk on the Gaussian sphere.

Both [44] and [34] assume that the image is aligned with the 3D model. Our method is a big step forward in this respect because it could recognize objects under arbitrary pose and lighting direction.

### *B. Detecting Specular Highlights*

Our approach uses specular highlights as its input; this requires a method for separating the specular component of the image. Much work has been done on separating highlights given multiple images, but in our case, a single-image method is needed. The method proposed in [17] first segments the image into color regions, and transforms the color of the regions into a color space that is invariant to the spectrum of the illuminant. For each region, a least-square approximation of the color values to a line in the 3D color space is performed, and then for each adjacent pair of regions the color lines are compared. If the lines intersect, then the region whose most color values lie beyond the point of intersection is labeled as a candidate highlight.

Their method is limited to cases in which the color spectrum of the illuminant and that of the object are different.

Klinker et al. [24] use a dichromatic reflection model which describes the color of every pixel as a linear combination of object color and highlight color. They project the color pixels into the color cube and then use PCA to approximate the projection by a plane. Their model predicts a T-shape of the color on this plane where one line corresponds to object color and a perpendicular line corresponds to the highlight color. They use an iterative process to separate the two lines and thus separate the highlight color component. Their method is limited to dielectric (non conductive) materials and also requires that the light and object colors are different.

In [50] the scene is captured with a polarizer at two linearly independent orientations. Then the 2-D polarization space is formed by the two sampled images. The diffuse and specular component can be separated in this space, since the polarization state of the diffuse component is different from that of the specular components. The advantage of this method is that it works for both dielectric and metal objects, and that it works regardless of the object and light colors. The disadvantage is that it requires the images to be taken in a controlled setting.

Ortiz and Torres [33] create an Intensity-Saturation histogram of the image which is generated from the Intensity-Saturation-Value color space. Then they detect the highlights as pixels of high intensity and low saturation. They use a local contrast enhancement algorithm in a preprocessing stage in order to normalize the dynamic range of the input.

In our tests a relatively simple two-step threshold on the intensity of the input pixels worked quite well. However, the discussed methods should be considered by a practitioner in tailoring the proposed recognition approach for a specific application.

### *C. Finding Correspondences between the Image and the Model*

In our task, we have a database of 3D models. We are given an image which is acquired by translating and rotating one of the models in 3D-space, and projecting the result into a plane. Therefore, by determining the pose, one means finding the 6 parameters of the 3D translation and rotation which align the projection of the model with the image. One way of finding these parameters is by using correspondence pairs between the image and the 3D model which involves the following steps: 1) detecting prominent features, 2) finding correspondences using these features, and 3) finding the pose from correspondences.

Features are detectable and well-defined parts of the image and the model. Correspondences, in this context, can be defined as pairs of features, one from the 2D image and one from the 3D model, which share common properties. The simplest form of correspondences are point pairs  $(P_I, P_M)$ , where  $P_I$  is a point in the image space,  $P_M$  is a point in the model space, and both refer to the same point of the object. More complex examples are pairs of lines and pairs of regions. An important problem in pose estimation is detecting the features, and then, finding the correspondence between them. There exist many approaches for detecting features and finding correspondences in the 2D-3D setting.

Schmid et al. [42] review and evaluate different feature detectors. They divide the features into three types:

- Contour-based features: corners, junctions, endings, straight lines, high curvature points.
- Intensity-based features: points which "stand out" on their background, regions of similar color or brightness, extrema of color derivatives and certain values of second derivatives.
- Parametric models, which fit parametric intensity model to the signal.

The features that we use in this work are of a completely different nature. We use photometric features, namely the centroids of specular highlights.

#### *D. Pose from Correspondences*

The problem of finding the pose from correspondences is widely studied in the literature. Here, we use correspondences obtained from highlights produced by an object in the input image and the number of highlights is usually small. Thus we need a method that finds pose from point correspondences and uses the minimum number of points possible. Haralick et al. [19] review solutions to the problem of finding the 6-parameter pose under perspective projection, using correspondence between three pairs of points. Three pairs of points is the minimum information which leads to a direct, closed-form solution to this problem. The direct three-points solution has a 4-fold ambiguity in this case, meaning that up to 4 possible poses will satisfy the correspondences. Over the years, approximations of the exact solution were proposed. Alter [1] considers an approximation to the perspective projection, in order to simplify the computations involved in computing the pose. He uses the weak perspective projection, which is equivalent to an orthographic projection, followed by scaling of the entire scene. This approximation works well in cases where the depth of the entire scene is small relative to the distance from the camera.

The advantages of using this approximation are lower computational cost, and reduction of the ambiguity of the solution to a 2-fold.

We employ the method from [1] for computation of the hypothesized poses as it satisfies our requirements and it is very fast to compute. The weak perspective assumption used in this method holds in our settings.

### III. THE BASIC APPROACH

Our method attempts to recognize the object viewed in the scene, assuming it is one of the models in the predefined database. Other outputs of the system are the pose of the object and the direction of light in the input image. The general scheme follows the method of Huttenlocher and Ullman [21], in which the objects to be recognized are first aligned with the image by finding the transformation (pose) of the 3D object and then verified by comparing the aligned models with the original image. The object with the highest verification score is the output of the recognition system.

Here, we compute hypothesized pose of the object from point correspondences as in [1]. The correspondences are obtained from highlights produced by the object, using an efficient algorithm (Section III-B) which builds upon the invariant properties of highlights, introduced in the next section. The aligning step is followed by the verification, which measures the similarity between the specular highlights extracted from the input image and the highlights predicted for the hypothesized pose, using a simple model of highlight formation proposed in [34].

#### A. Geometric Model

We want to match a highlight in an image to its corresponding 3D patch. A naive approach is to render highlights for all possible lighting and viewing directions and then apply a brute force search to find the match. Even before considering the scale and rotation, this approach requires rendering  $T^2$  highlights, where  $T$  is the number of points in the tessellation of a viewing sphere. After adding the scale and rotation, it becomes intractable. The following observation allows to reduce drastically the hypothesis space, which improves both the complexity and accuracy of the matching.

We introduce the basic idea on a simplified case and then we show how the same concept can be applied to general objects under certain assumptions.

**Proposition 1.** Assume that the scene is illuminated by a single, distant, compact light source and the distance to the camera is large enough to assume weak perspective projection. A planar, mirror patch with normal  $\vec{N}$  will appear specular in all images with viewing direction  $\vec{V}$  and lighting direction  $\vec{L}$ , such that  $\vec{N}$  is a bisector of the angle between  $\vec{V}$  and  $\vec{L}$  (and  $\vec{N} \cdot \vec{L} > 0$ ). Further, the shape of the highlights in all these images is approximately the same up to affine transformation.

**Proof.** According to the law of reflection, a point with a normal  $\vec{N}$  will appear specular in all images with viewing direction  $\vec{V}$  and lighting direction  $\vec{L}$ , such that  $\vec{N}$  is a bisector of the angle between  $\vec{V}$  and  $\vec{L}$  (and  $\vec{N} \cdot \vec{L} > 0$ ). Since all points in the planar patch have the same normal  $\vec{N}$ , the whole patch will appear specular in all these images. In other words each such image is a different projection of the same planar patch, and the patch is highlighted in all of them (as we just showed). Since different 2D orthogonal projections of a planar patch are related by an affine transformation, the shape of the highlights corresponding to the planar patch in all these images is the same up to affine transformation (assuming weak perspective projection).

Next we generalize these observations to a 3D specular object with non-mirror reflectance.

**Proposition 2.** Assume a 3D smooth object, with a uniform, specular, non-mirror reflectance, illuminated by a single, distant, compact light source of an arbitrary direction  $\vec{L}$ . Assume that  $\vec{N}$  is a surface normal that produces the highest specular reflection. Denote the set of highlighted surface normals as

$$\text{spec}(\vec{N}) = \left\{ \vec{N}' \mid I(\vec{N}') \geq t \right\}$$

where  $I(\vec{N}')$  denotes the specular component of the intensity at the point with the surface normal  $\vec{N}'$  and  $t$  is a constant threshold. Given  $\vec{N}$ ,  $\text{spec}(\vec{N})$  doesn't depend on  $\vec{L}$ .

We use the Blinn-Phong model [8] to describe the specular component of the intensity of points in  $\text{spec}(\vec{N})$ . According to this model,

$$I(\vec{N}') \propto (\vec{H} \cdot \vec{N}')^\alpha,$$

where  $\alpha$  is an exponent that depends on the glossiness of the surface and  $\vec{H} = \frac{\vec{L} + \vec{V}}{\|\vec{L} + \vec{V}\|}$  is a vector in the direction halfway between  $\vec{L}$  and  $\vec{V}$ .

Since  $\vec{N}$  is the surface normal that produces the highest specular reflection,  $\vec{N} = \vec{H}$ . Thus

under the Blinn-Phong model,

$$\text{spec}(\vec{N}) = \left\{ \vec{N}' \mid (\vec{N} \cdot \vec{N}')^\alpha \geq t' \right\}$$

which doesn't depend on  $\vec{L}$  ( $t'$  absorbs the additional constants).

According to Proposition 2, changing the lighting and viewing directions such that  $\vec{N}$  remains to be the normal with the highest specular reflection, doesn't change  $\text{spec}(\vec{N})$ . Note that for the high values of  $t$  (which is usually the case with the glossy objects), normals in  $\text{spec}(\vec{N})$  are very similar and thus the corresponding surface patch (patches) is (are) approximately planar. Therefore, different orthogonal projections of a highlight are related by an approximately affine transformation.

Now, instead of rendering all possible highlights, produced by a surface patch (as would be done in the naive approach) we can render a single image, corresponding to  $\vec{L} = \vec{V} = \vec{N}$  (we choose this particular direction, since it minimizes the distortion) and use it as a reference for all images in which this surface patch appears specular. Further, representing the highlights by their affine invariant descriptors provides invariance to the transformations resulted from a change in viewing direction, and to rotation and scale. This way we reduce the search for the best match to linear in the tessellation size.

### *B. Finding Correspondences using Highlights*

The centroid of a planar 3D region is viewpoint-invariant under weak perspective projection, thus one can assume that the centroid of the highlight and the centroid of the 3D patch that produced the highlight are approximately the same. Consequently, one can use its 2D coordinates in the image and its 3D coordinates in the model as a correspondence pair. Having three such pairs is theoretically enough to compute the pose parameters (up to 2-fold ambiguity). Since only a single point – the centroid – is taken from each highlight for establishing the correspondence, at least three highlights in the image are needed for finding the pose solely from specular highlights. However, if other cues are available (prominent shape or texture features), one can easily integrate the correspondences obtained from different sources to find the candidate pose. Later on we show that the verification of the pose uses only the highlights and it has no limitation on the number of highlights (even a single highlight could suffice for the verification).

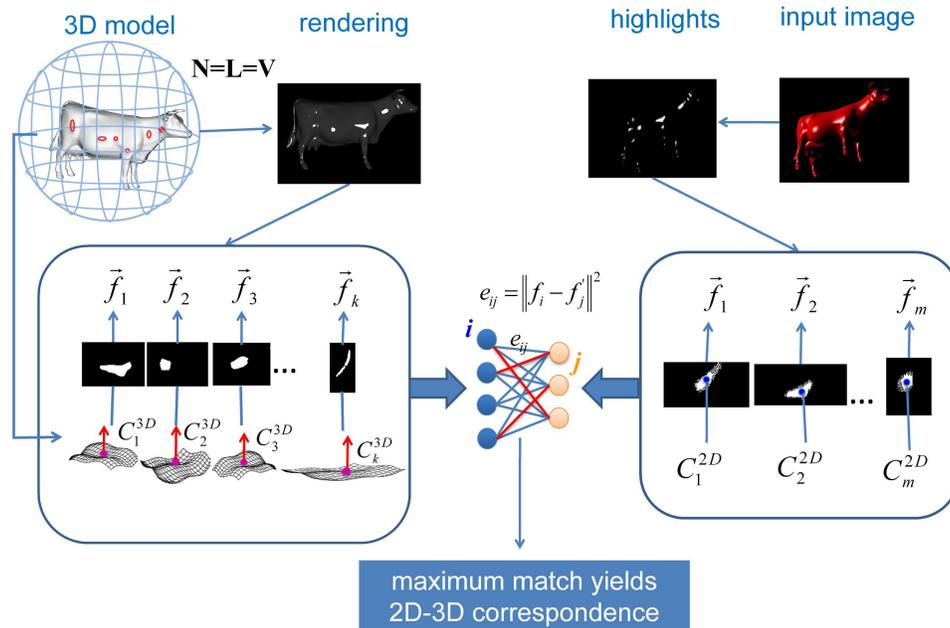


Fig. 2. Schematic overview of 2D-3D correspondence.  $C_i^{3D}$  denotes the 3D centroid of a surface patch that produces a highlight,  $C_i^{2D}$  denotes the 2D centroid of a highlight in the input image,  $f_i$  denotes an affine invariant descriptor of a highlight,  $N$  is the tessellation point on the viewing sphere,  $L$  is the lighting and  $V$  is the viewing direction in the rendered image.

An efficient way of matching between a highlight in an image with unknown lighting and pose and a 3D patch that produced it, includes an offline and online stages.

During the offline stage, we render the highlights as viewed from each point on the viewing sphere (according to some tessellation) for the special case in which the lighting direction  $\vec{L}$  coincides with the viewing direction  $\vec{V}$ . This is done by first identifying the specular normals as  $\{\vec{N}' | \vec{N}' \cdot \vec{N} > \tilde{t}\}$ , where  $\vec{N} = \vec{L} = \vec{V}$  and  $\tilde{t} = t^{1/\alpha}$  (see the proof of the Proposition 2), and then rendering these normals as specular. We find the highlights in the rendered image as the connected components of white pixels. For each highlight we compute an affine invariant descriptor and the 3D centroid of the the surface points that produced it (for each specular pixel we know the 3D point that generated it) and we store the descriptors and the 3D centroids for all highlights in the view, marked by the viewing direction.

The online stage consists of extracting the highlights from the input image, calculating the affine invariant descriptor and the 2D centroid of each significant highlight, and then matching the invariants in the given image to the precomputed invariants of the rendered highlights in the stored views.

Matching all highlights that appear in a view as a set, as opposed to matching each highlight

individually, has its positive and negative aspects. The advantage is that matching a set of highlights has a lower chance of false matches compared to matching an individual highlight and it's computationally more efficient. On the other hand, a rendered view and an input image could have different number of highlights due to self occlusions. The following matching procedure allows for unmatched highlights. A schematic overview of the matching is shown in Figure 2. For each viewing direction according to the tessellation, we construct a full bi-partite graph in which one side corresponds to the highlights in the input image and the other to the rendered highlights stored for that view. The weights on the edges are the Euclidian distances between the affine invariant descriptors of the highlights, represented by vectors (Section III-C). Note that the highlights in the real and the rendered images are not the same, but they are related by an affine transformation (see Proposition 2), thus affine invariant descriptors of the highlights should resolve the difference up to noise. We use Hungarian algorithm [26] to find the best maximum match for the bi-partite graph. Matching also relates between the 2D centroids of the highlights in the input image and the 3D centroids of the surface points (these have been stored with the view), which provides the 2D-3D correspondences needed for the pose estimation algorithm. The number of highlights in every view is usually small, which makes the matching very fast.

The proposed matching process chooses a subset of views, which we call candidate views, that best match the invariants computed from the input image. Only the candidate views are used for pose estimation, and verification. The exact number of candidate views needed for correct pose estimation depends on the object. If the object has a very complex shape and most of its local parts differ from one another, the highlights are distinctive enough and the number of candidates could be rather small. For more symmetric objects, the number of candidate matches could be high. Nevertheless, the method remains efficient: first, because sets of highlights are matched instead of individual highlights, which decreases the number of false matches; second, for every candidate view, polynomial matching is applied in order to establish correspondence between the highlights in this view to the highlights in the image, and up to two hypothesized poses<sup>1</sup> are computed for the view, as opposed to computing poses for all possible correspondences of the highlights.

<sup>1</sup>Two poses are computed due to the 2-fold ambiguity of the 3-point pose estimation algorithm.

### C. Affine Invariants

We choose Affine Moment Invariants as descriptors of highlights [14, 46] due to computational efficiency and low storage requirements. Given a binary image of a highlight, cropped from the image of the object, we compute 34 independent invariants up to weight 10 polynomials [46] as in the central moments of the image and combine them into a single vector, which we use as an affine invariant descriptor of the highlight.

Computing and storing affine invariant descriptors for every direction on the viewing sphere is the most computationally expensive part of the proposed method. Fortunately, it can be done in an offline phase.

### D. Pose from Correspondences

Specular highlights produced by a single light source are sparse for most real objects, thus we need a method for pose computation that works with a minimal number of correspondences. We choose [1] as it needs only three correspondences and it is computationally efficient. This algorithm, however, has 2-fold ambiguity of the solution, thus when the image contains three highlights, two hypothesized poses are computed per view. In practice, there are not many cases with more than three significant highlights. In such cases, the same algorithm is applied for all possible triplets of correspondences, and the correspondence which gives the lowest error on the rest of the points is then used as hypothesized pose of the view.

### E. Verification

The purpose of the verification step is to quantify the resemblance between the highlights extracted from an input image and the highlights predicted for a hypothesized pose and identity of the object. The hypothesized pose allows one to match image pixels to corresponding surface normals on the model. We map each highlighted pixel from the input image to a point on a Gaussian sphere having the same surface normal and label this point as specular. According to the model introduced in [34], if the pose and the model are chosen correctly, the normals corresponding to the specular pixels must form a cap on a Gaussian sphere. The size of the cap is determined by the material properties of the object. Since these are known (or can be estimated), we could adjust the labelling on the sphere such that the specular normals form a cap of the correct size. The size of the cap can be controlled by  $\tilde{t}$ , which is the threshold on the dot

product between the central normal and the most peripheral normal within the cap. In practice, we search for a normal  $\vec{N}$ , for which the set of specular normals  $\{\vec{N}' \mid \vec{N}' \cdot \vec{N} > \tilde{t}\}$  is the largest.  $\vec{N}$  is chosen to be the center of the cap and the set of all normals  $\vec{N}'$  satisfying  $\vec{N}' \cdot \vec{N} > \tilde{t}$  to be specular. The updated labelling is then mapped back to the image plane and compared with the original highlights. This process relies on the assumption that if the hypothesized pose and identity are correct, the updated highlights will be similar to the original, but if the hypothesized pose or identity are wrong, the updated highlights will be inconsistent with the original (Figure 3).

The overlap measure used in Osadchy et al. [34] is defined as

$$overlap = \frac{size(B_I \wedge B'_I)}{size(B_I \vee B'_I)} \quad (1)$$

where  $B_I$  is the binary image of the original highlights and  $B'_I$  is the binary image of highlights mapped back from the Gaussian sphere. We found that this measure is not robust to small shifts, caused by the errors in pose. To overcome this, a robust variant of Hausdorff distance [45] is applied to compare the binary images:

$$H(B_I, B'_I) = h(B_I, B'_I) + h(B'_I, B_I),$$

$$h(A, B) = \frac{1}{|A|} \sum_{a \in A} \min\{\alpha, \min_{b \in B} \|a - b\|\}$$

where  $|A|$  is the number of non-zero pixels in  $A$ ,  $\|a - b\|$  is the normalized Euclidean distance between points, and  $\alpha$  is a smoothing constant that determines the balance between stability and discriminative ability.  $\alpha = 0$  means that the distance will always be zero, and  $\alpha = 1$  means no smoothing. In our experiments we used  $\alpha = 0.05$ .

We found that verification score function has a steep slope near the optimum pose. Thus we can further improve the accuracy of pose estimation by running optimization of the verification function, with the hypothesized pose as a starting point. To this end we choose a few hypothesized poses with the best verification score and run a standard routine for constrained non-linear optimization [35] using these poses as starting points. The pose that produces the best score (after optimization) is chosen to be the pose of the object. If the model used in the process is the correct one, then this pose is the true pose of the object, otherwise the pose has no meaning.

The verification process allows to recover the light direction in the scene. After adjusting the

specular cap on the sphere, its central normal  $\vec{N}$  corresponds to the bisector between the light direction  $\vec{L}$  and the viewing direction  $\vec{V}$  (which is (0,0,1) in the camera centered coordinate system). Note the object's normals are rotated using the estimated rotation. The light direction  $\vec{L}$  can be estimated as

$$\vec{L} = 2(\vec{N} \cdot \vec{V})\vec{N} - \vec{V}.$$

#### IV. THE RECOGNITION SYSTEM

Next we present an example of a system for recognition of specular free-form objects. The system incorporates the concepts introduced in the previous section with other modules, the implementation of which depends on the application at hand. The algorithm is quite efficient and most of its steps can be parallelized, which allows close to real time performance.

##### A. Offline Stage

The offline stage performs the following steps for every object in the database.

Given a 3D model of an object, define a set of unit vectors  $\{\vec{N}_i\}$  which is a subset of the object's normals according to a certain tessellation. For each  $\vec{N}_i$  perform the following steps:

**Step 1.** Set  $\vec{V} = \vec{L} = \vec{N}_i$  where  $\vec{V}$  is the viewing direction and  $\vec{L}$  is the lighting direction. Render a binary image  $B_i$  of the object from the viewing direction  $\vec{V}$  according to the model introduced above: the intensity of a pixel is set to one if the dot product between its corresponding normal and  $\vec{N}_i$  is larger than a predefined shininess threshold  $\tilde{t}$ , otherwise the intensity is set to zero.

**Step 2.** Locate significant highlights in  $B_i$  by finding the connected components and removing very small ones. Small shapes are more prone to error in the affine invariants since they consist of a small number of discrete pixels.

**Step 3.** Compute an affine invariant descriptor (see Section III-C) for every significant highlight in  $B_i$ . Store the descriptor along with the 3D centroid of the surface patch that produced it and with the normal  $\vec{N}_i$ .

##### B. Online Stage

Given an image  $I$  of an unknown object perform the following steps:

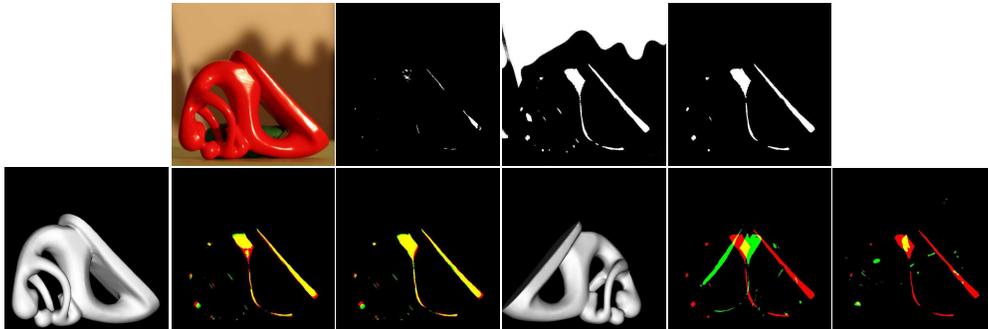


Fig. 3. Step 1 and 5 of the online stage. Top row, left to right: a given image with unknown pose and lighting, high threshold binary image, low threshold binary image, highlights from low threshold which intersect with those that passed the high threshold; Bottom row, left to right: correct hypothesized pose, initial overlap for correct pose (threshold highlights are in red, rendered highlights are in green, the overlap between the two is in yellow), refined overlap (using the Gaussian sphere), incorrect hypothesized pose and the corresponding overlap, overlap with an incorrect hypothesized model (cane). This figure is best viewed in color.

**Step 1.** Extract the specular highlights in  $I$  by thresholding the specular component of the image. Let  $B_I$  denote the binary image of the extracted highlights. The significant highlights in  $B_I$  are determined in the same way as in step 2 in Section IV-A.

Section II-B discusses different methods for separating the specular and diffuse components of the image. In our experiments we used a simple two-step threshold on the intensity of the input pixels to extract the specularities, which first applies a high threshold on the image, and then a low threshold but selects only the highlights which intersect with those that passed the high threshold (Figure 3).

**Step 2.** Compute the affine invariant descriptor (see Section III-C) and the 2D centroid for each significant highlight in  $B_I$ . Now  $I$  is represented by a set of  $(centroid, descriptor)$  pairs. The size of the set is equal to the number of significant highlights in  $B_I$ .

For each 3D model  $M_k$  in the data base apply steps 3 to 6:

**Step 3.** Find candidate correspondences between the highlights in  $I$  and the 3D model  $M_k$  and rank them. To this end, for each viewing direction according to the tessellation defined in the offline stage, construct a full bipartite graph as shown in Section III-B and find the best maximum match using Hungarian algorithm [26]. The maximum match relates between the 3D centroids of the highlights, generated for the 3D model in the offline stage, and the 2D centroids of the highlights that were extracted from the input image. Assign a score to the found correspondence as the average distance between the descriptors of the matched highlights. Choose up to  $K$  views, with matching

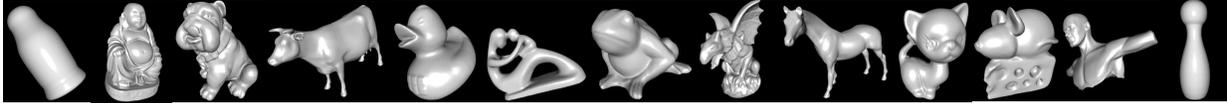


Fig. 4. Models used in the experiments: doll, Buddha, cane, cow, duck, fertility, frog, gargoyle, horse, kitten, mouse, torso, pin. These names are used in the following figures.

score higher than a predefined threshold, as candidates views. (Alternatively, a given percentage of views with the highest score can be chosen.)The threshold and  $K$  are chosen empirically.

**Step 4.** For each candidate correspondence, compute the pose as described in Section III-D. Discard poses with an out-of-bounds translation vector. In the case of more than 3 points, each triplet produces a pose. A pose which is inconsistent with poses obtained from other triplets is likely to be erroneous and should be discarded.

**Step 5.** For each pose, obtained in Step 4. apply the verification process as described in Section III-E.

**Step 6.** (Optional) Choose  $S$  poses with the highest verification score and use them as starting points in the optimization of verification function <sup>2</sup>. Choose the pose that produces the best score (after optimization). Store the verification score for the model  $M_k$  and go back to Step 3.

**Step 7.** Identify the object in  $I$  as the one corresponding to the model with the highest verification score.

## V. EXPERIMENTS

We built a new data base for model-based recognition of specular objects and ran all the experiments on this set. The data base is available at <http://www.cs.haifa.ac.il/~rita/specObj/main.html>. It contains 3D models of 13 objects and images of these objects under varying viewing conditions.

The 3D models of 11 objects (2 – 12 in Figure 4) were obtained from AIM@SHAPE Shape Repository (<http://shapes.aim-at-shape.net/>) and used to create real objects using 3D printing technology which allows to produce objects from a CAD model with relatively high accuracy. These objects were painted with a glossy paint, which produces specular effects (Figure 5). We

<sup>2</sup>due to running time constraints we ran optimization on very few poses ( $S = 3$ ).

colored the objects with the same uniform color, since textureless objects are more challenging for pose estimation and recognition in general. Our method gains no advantage from the uniform color and doesn't make any assumptions about the texture of the objects. However, to show that it can work with textured objects, we added two such objects to the data base (Figure 5). The 3D models of these objects (doll and pin in Figure 4) were computed from 2D images assuming rotational symmetry [34]. The 3D models of all objects were centered, bound to the unit sphere in size, and remeshed to have between 50,000 and 100,000 faces. Processing of the models was done using MeshLab (<http://meshlab.sourceforge.net/>).

The data base contains 741 images of 13 objects, divided into three subsets: synthetic, real outdoor, and real indoor. The images contain variations in lighting conditions (direction, intensity, and the size of the source), background, pose, and size of the objects. Some of the images contain large shadows and partial occlusions. The data base also includes images in which the object is slightly out-of-focus, which tests the robustness of the affine invariant matching. The examples of real images with some of the listed variations are shown in Figures 5 and 9.

**Synthetic Images:** For each object, we have generated 20 synthetic images in random poses with random light direction, restricting them to have at least three highlights (in total 260 images). Figure 4 shows examples of the synthetic images with various levels of shininess.

**Outdoor Set:** The outdoor subset contains 155 images against black background and 76 images against cluttered background, in total 231 images. All the images were taken with the sun light in different hours of the day. Thus some variation in lighting direction is present in these images (due to sun movement).

**Indoor Set:** The indoor subset contains 250 images against both plain and cluttered backgrounds and includes large variation in illumination direction. In many of these inputs, the light source and camera were not very far from the object, which allowed us to test the robustness of the algorithm against deviation from the assumption of a distant light source and a weak perspective projection. Some of the images include partially occluded objects.

### A. Implementation Details

The highlights in the offline stage of the algorithm and all images in the synthetic set, were rendered at a resolution of  $1024 \times 1024$ . The resolution of real images varied from 866 to 2053 (all images had square size). For the verification step (Section III-E), rendering of the mapped-



Fig. 5. Viewing conditions in the test sets. First row: examples of the lighting conditions – the first four images were captured in the sun light, the next three images were taken with an indoor light source with varying direction. Second row: the first four images show the size variation, the next three images provide examples of partial occlusions. Both rows show typical backgrounds in our experiments.

back highlights was done at a resolution of  $256 \times 256$ . The number of candidate views,  $K$ , chosen for verification was 20% for the synthetic images and 50% for the real images.

The algorithm was implemented mostly in MATLAB and partially in Java for the OpenGL renderings. The average (online) running time of the algorithm was around 60 seconds per input image, which could be significantly improved by a more efficient implementation and also by parallelization, which is possible during most stages of the algorithm. The test runs were performed on a laptop with Intel Core i7 Q 720 CPU and NVIDIA GeForce GTS 250M graphics card.

### B. Determining $\tilde{t}$

In order to find  $\tilde{t}$ , we use a 3D model of the object and its image aligned with the model. After segmenting the highlights, we map the pixels within the highlights to the points on the Gaussian sphere having the same surface normal. According to the model, introduced in Osadchy et al. [34], the specular points of the sphere must form a cap, which we find using the algorithm from Osadchy et al. [34]. We set  $\tilde{t}$  to the value of the dot product between the normal in the center of the cap and the most peripheral normal within the cap.

### C. Recognition Results

We ran recognition tests separately for synthetic, indoor, and outdoor sets. The results are summarized as confusion matrices and shown in Figure 6. Recognition rates of objects (the percentage of images of the object that were recognized correctly) are listed along the diagonal.

The average recognition rates of the synthetic, outdoor, and indoor sets are 87.3%, 77%, and 72.5% correspondingly. The gap in the performance between the synthetic and the real sets results from noisier specularities extraction and violation of assumptions in real images. The average recognition rate on the indoor set is the lowest, because the positions of the light source and the camera were close to the object in at least half of the indoor images, and the indoor set includes images with partially occluded objects. Section V-E provides a detailed discussion about the sensitivity of our method.

To improve the robustness of recognition, one could consider rejecting inputs with confidence level lower than a certain threshold. Inputs with low confidence level could come from objects that are not in the database or images with high level of noise. Such confidence measure can be easily obtained using the normalized distance  $H$  employed in the verification step (Section III-E).  $H = 0$  characterizes a perfect match between the highlights, and  $H = 1$  means no match. Therefore, we can use  $1 - H$  as the confidence measure of the result. Figure 6 (bottom, right) shows the average recognition rates as a function of rejection rates for the three sets. We can see that by rejecting a small fraction of the inputs with low confidence, we improve the recognition results and reliability of the system.

#### *D. Pose Accuracy Validation*

Measuring the accuracy of pose is not trivial both mathematically and practically. First, because it depends on application and second, because it requires accurately labeled poses and these are very hard to obtain (it either requires a robotic arm or manual labeling which is extremely tedious).

Given the true pose  $T$  and the pose found by the algorithm  $\tilde{T}$ , a measure of distance between  $T$  and  $\tilde{T}$  is required. Our pose space is essentially 6-dimensional: 3 dimensions of translation, and 3 dimensions of rotation. Measuring distance between the 3 translation parameters is straightforward, using the Euclidean distance between the translation vectors. Measuring the rotation distance is more difficult, and depends on the representation of the rotation.

A popular representation of rotation is the Euler angles. They are the most compact representation, but there is no straightforward way of denoting distances between two triplets of angles as a single number. In our work we used the axis-angle representation, which is described by two parameters: a unit vector indicating the direction of the rotation axis and a real value describing

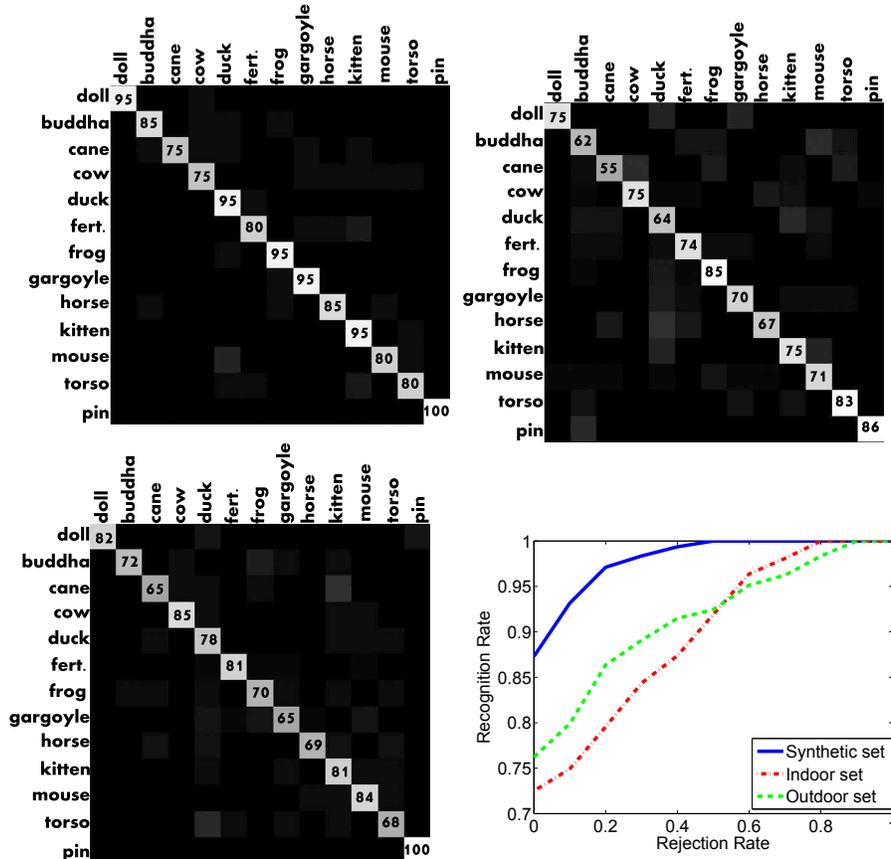


Fig. 6. Recognition Results. Confusion matrices: top left, indoor set; top right, outdoor set, bottom left, synthetic set. Average recognition rates of objects are listed along the diagonal. Bottom right, average recognition rates for the three sets as a function of rejection rate.

the angle of rotation about this axis.

We have evaluated the accuracy of the pose estimation separately for translation error and rotation error. Denote the true translation vector as  $\tau$  and rotation matrix that corresponds to the true rotation angles as  $R$ . Denote the corresponding output of the algorithm as  $\tilde{\tau}$  and  $\tilde{R}$ . The translation error is defined as  $\|\tau - \tilde{\tau}\|$ . The rotation error is defined as the angle that corresponds to the axis-angle representation of the rotation matrix that brings from  $R$  to  $\tilde{R}$ . Since we do not use the texture of the object, pose estimation of a rotationally symmetric object is ambiguous around the axis of symmetry. Thus for such objects, we measured the rotation error as an angle between the true and the estimated axes of symmetry.

We used the correctly recognized images in the synthetic set and in a subset of the outdoor set, which were manually labeled for the 2D-3D correspondences (165 images), to evaluate the translation and rotation errors of the pose estimation algorithm. Figure 7 shows the rotation

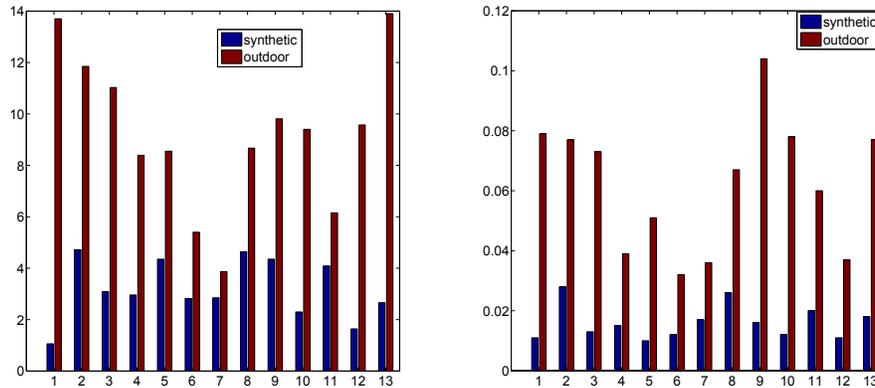


Fig. 7. Pose estimation errors in the synthetic set (including 260 images) and in a subset of the outdoor set (including 165 images with pose annotation obtained from manual correspondences). Left – rotation error, right – translation error. Rotation errors are given in degrees, and translation errors are relative to the object size. The x-axis enumerates the models according to the order given in Figure 4.



Fig. 8. Examples of pose estimation results on real images. The white contour corresponds to the occluding contour of the object in the estimated pose. 5 last images in the bottom row show some of the failure cases. Note that the shape and the location of the highlights are plausible for the incorrectly estimated poses, which can explain the errors.

and translation errors for the two sets. Due to the high difficulty of the manual correspondence labeling, we did not annotate the entire real set. However, the success of the pose estimation can be judged by the recognition success. Figures 8 and 9 show examples of real images and illustrate the result of the pose estimation.

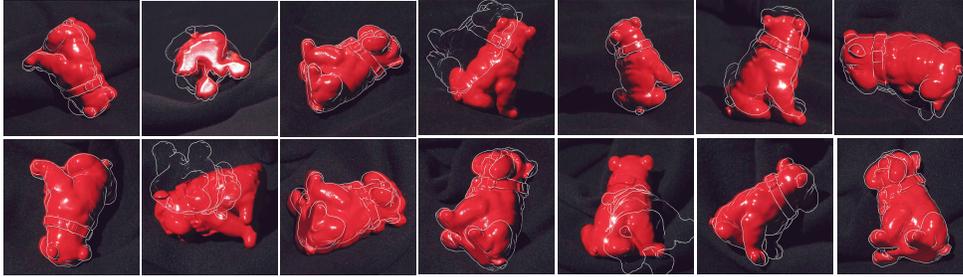


Fig. 9. Variation of poses for one of the objects from the outdoor subset. The pose estimations shown by white contours are overlaid with the original images.

### *E. Robustness*

Our experiments contain images with large variation in light source and viewing directions, showing the robustness of the proposed approach to these factors. The real images are illuminated by sun and by lamps of different sizes and intensities, which shows the robustness of our method to the intensity and color of the light source.

In some of the tested images the object of interest was partially occluded either by the holding hand or intentionally (see Figure 5, bottom row for examples). Since the highlights are local and sparse, the chance of occluding them is low. Thus our method is quite robust to partial occlusions. 44 images in the indoor set contain partial occlusions; 31 of them were recognized correctly, which is 70.5% compared to 72.5% for the entire indoor set.

The proposed approach is insensitive to cluttered background as long as it doesn't exhibit highlights. In our experiments we used several types of background (cluttered and homogenous); Figure 5 shows the variation in background in our tests. Some of the images have background highlights. When their shapes and positions don't match any of the objects' highlights, our method performs well. However, we cannot claim the robustness to background highlights in general. We plan to address this problem in future research.

In order to test the robustness of the recognition algorithm to object size, we took 8 scenes of four objects that were recognized correctly in previous experiments and created 5 croppings per scene with varying amount of background. Then we scaled the images to the same size. This way we vary both the resolution of the object and its size relative to the scene. The upper limit on the size of the object is constrained by the requirement that the object should fit the image. The smallest size is not defined, but we set the lower limit on the number of pixels, comprising a highlight. The plot showing the recognition rate as a function of object size (averaged over

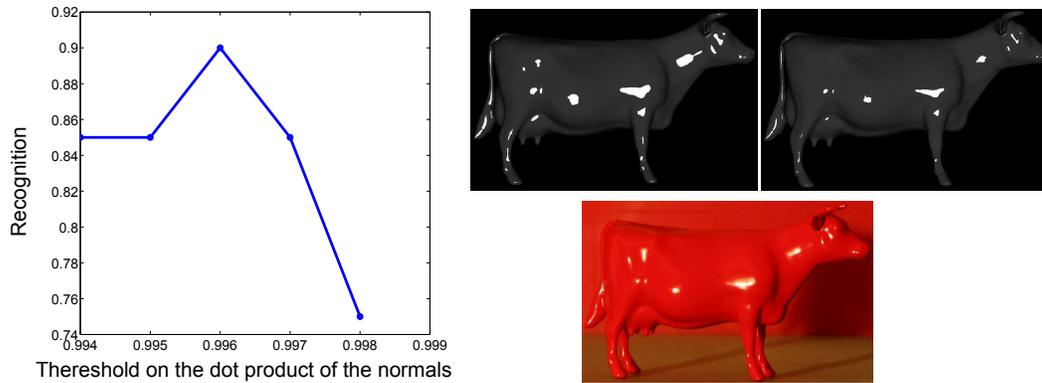


Fig. 10. Sensitivity to errors in the estimation of the specular properties. Left – recognition rate as a function of  $\tilde{t}$ . Right – real image and binary renderings with the maximum and minimum of the tested values of  $\tilde{t}$ .

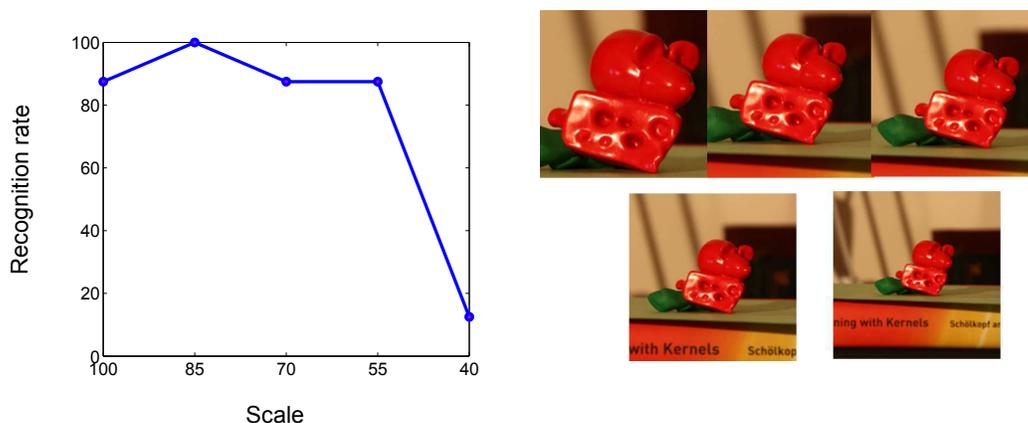


Fig. 11. Sensitivity to the size of the object relative to the size of the scene; left – recognition rate as a function of object scale; right – examples of images with scales used in the test.

the scenes) and an example of croppings for one scene are shown in Figure 11.

We assume that the object of interest has uniform reflectance. However, in practice, the paint has small variations in reflectance properties and even finger marks left on the objects from holding them change their reflectance, but our method appears to be robust to such variation. To test the sensitivity of the algorithm to errors in the estimation of specular properties of the object, which are characterized by  $\tilde{t}$  (the threshold on the dot product of the normals defining the highlight size), we ran the recognition algorithm on 20 images of two objects using 5 values of  $\tilde{t}$ . Figure 10 shows the recognition rate as a function of  $\tilde{t}$  and the renderings obtained with min and max values of  $\tilde{t}$ . This test shows that our method is moderately robust to the errors in estimation of the material properties.

## VI. DISCUSSION AND FUTURE WORK

In this work we addressed a challenging task of recognition in difficult viewing conditions, in which conventional features for establishing 2D-3D correspondence are unreliable. We showed that for shiny objects, under the assumption of a dominant light source, specular highlights could be used as pose-invariant features. We developed a pose estimation and recognition algorithms that rely solely on highlights and do not require the knowledge of lighting. The proposed method showed good results in an extensive evaluation that included both synthetic and real images of complex objects. We provided a data base of complex shiny objects that includes accurate CAD model of the objects and their real images under varying viewing conditions.

The two main sources of errors in our method are incorrect extraction of highlights and similarities in the local shape of the object. The specularity extraction is a non-trivial task and there is no generic solution for it yet. Previous work shows nice results for assumed conditions (Section II-B), thus the choice of the method for highlight extraction should be carefully considered based on the application at hand.

Different parts of a symmetric object could produce highlights with the same shape up to affine transformation. Our current approach matches the highlights in the input image to the rendered view using the Euclidian distance between the affine invariants of the highlights and only the best match is considered. Thus if the correct view contains several highlights with similar shape (up to affine transformation) the best match is ambiguous and in such cases the current implementation is likely to fail. Previous work shows that there is a significant advantage in resolving ambiguities during the hypothesis testing instead of the matching phase [20]. The straightforward solution is to check all possible correspondences in the view, but this solution is exponential in the number of highlights. Moreover each such match should be validated, but our current validation requires a rendering step and thus is not very efficient. In future research we shall consider extending the current approach to several matches per view with more efficient validation that doesn't involve rendering. These will also help in developing more practical method that can handle background highlights, multiple light sources and objects in the scene.

There are other sources of errors, but they are much less common. For example, partial occlusion of a highlight could change the affine invariant descriptor, which results in a wrong match. Future work may address these cases by detecting abnormalities in the shapes of the

highlights and filtering those highlights. Objects with many ridges have patches with similar normals close to each other. This results in highlights which are close to each other, and for some viewing direction such highlights merge into one highlight. This could also confuse the matching procedure. For example, our method has lower recognition rates on the gargoyle object that has many such ridges.

The reflectance properties of the objects are assumed to be part of the object's model. If these parameters are not given, we showed how to estimate them. Objects with the same shape but different reflectance could be considered as different reference models. It is possible to generalize the proposed algorithm to varying reflectance properties by optimizing the verification function over the specular cap size (Section III-E).

There are parts of the algorithm that could be further optimized, for instance, the search of the best matching views is linear in the number of samples on the viewing sphere. Reducing the number of viewing directions could result in errors in pose estimation. A possible solution is to use non-uniform tessellation, which is sparser on smooth parts of the object and denser in areas of high curvature. This and other optimizations will be explored in future work.

## REFERENCES

- [1] T.D. Alter, Massachusetts Institute of Technology, and Artificial Intelligence Laboratory. *3d pose from 3 corresponding points under weak-perspective projection*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1992.
- [2] B. Barrois and C. Wohler. 3d pose estimation based on multiple monocular cues. In *BenCOS07*, pages 1–8, 2007.
- [3] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. on PAMI*, 25(2): 218–233, 2003.
- [4] J.S. Beis and D.G. Lowe. Indexing without invariants in 3d object recognition. *IEEE Trans. on PAMI*, 21(10):1000–1015, 1999.
- [5] J.W. Birk, R.B. Kelly, and H.A.S. Martines. An orienting robot for feeding workpieces stored in bins. *IEEE Trans. Systems Man Cybernet*, 11(2):151–160, 1981.
- [6] A. Blake and G. Brelstaff. Geometry from specularities. In *ICCV*, pages 394–403, 1988.
- [7] A. Blake and H. Bulthoff. Shape from specularities: Computation and psychophysics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 331(1260):237–252, 1991.
- [8] J. F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11: 192–198, 1977.

- [9] T. A. Cass. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In *ECCV*, pages 834–842, 1992.
- [10] J.Y. Chang, R. Raskar, and A. K. Agrawal. 3d pose estimation and segmentation using specular cues. In *CVPR*, pages 1706–1713, 2009.
- [11] A. Collet, M. Martinez, and S. S. Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30:1284 – 1306, 2011.
- [12] P. David and D. DeMenthon. Object recognition in high clutter images using line features. *ICCV*, 2:1581–1588, 2005.
- [13] A. DelPozo and S. Savarese. Detecting specular surfaces on natural images. In *CVPR*, volume 2, 2007.
- [14] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern recognition*, 26(1):167–174, 1993.
- [15] Y. Fukada, H. Doi, K. Nagamine, and T. Inari. ‘relationships-based recognition of structural industrial parts stacked in bin. *Robotica*, 2:147–154, 1984.
- [16] A. S. Georghiadis. Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In *ICCV*, pages 816–825, 2003.
- [17] R. Gershon, A.D. Jepson, and J.K. Tsotsos. The use of color in highlight identification. In *Proc. of the 10th int. joint conf. on Artificial intelligence*, volume 2, pages 752–754. Citeseer, 1987.
- [18] K.D. Gremban and K. Ikeuchi. Planning multiple observations for object recognition. *IJCV*, 12(2):137–172, 1994.
- [19] B.M. Haralick, C.N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 13(3):331–356, 1994.
- [20] E. Hsiao, A.Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2653–2660, 2010.
- [21] D.P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. ICCV*, volume 87, pages 102–111, 1987.
- [22] D. Jacobs and R. Basri. 3-d to 2-d pose determination with regions. *IJCV*, 34(2):123–145, 1999.
- [23] U. Klank, D. Pangercic, R. B. Rusu, and M. Beetz. Real-time cad model matching for mobile manipulation and grasping. In *9th IEEE-RAS Int. Conf. on Humanoid Robots*, December 7-10 2009.
- [24] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlights in color images. *IJCV*, 2(1):7–32, 1988. ISSN 0920-5691.
- [25] K. Koshikawa and Y. Shirai. A 3-d modeler for vision research. In *Proc. Inter. Conf. on Advanced Robotics*, pages pp.185–190, 1985.
- [26] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2: 83–97, 1955.

- [27] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua. 3d pose refinement from reflections. In *CVPR*, 2008.
- [28] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *ICCV*, pages 238–249, 1988.
- [29] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on PAMI*, 13(5):441–450, 1991.
- [30] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *CVPR*, volume 2, pages 973–979, 2005.
- [31] J.F. Norman, J.T. Todd, and G.A. Orban. Perception of three-dimensional shape from specular highlights, deformations of shading, and other types of visual information. *Psychological Science*, 15(8):565–570, 2004.
- [32] M. Oren and S.K. Nayar. A theory of specular surface geometry. *IJCV*, 24(2):105–124, 1997.
- [33] F. Ortiz and F. Torres. Automatic detection and elimination of specular reflectance in color images by means of MS diagram and vector connected filters. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans. on*, 36(5):681–687, 2006. ISSN 1094-6977.
- [34] M. Osadchy, D. Jacobs, R. Ramamoorthi, and D. Tucker. Using specularities in comparing 3D models and 2D images. *CVIU*, 111(3):275–294, 2008.
- [35] M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis, G.A.Watson ed., Lecture Notes in Mathematics*, 630, 1978.
- [36] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *JOSA*, 18(10):2448–2459, 2001.
- [37] S. Roth and M. J. Black. Specular flow and the recovery of surface structure. In *CVPR*, pages 1869–1876, 2006.
- [38] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition from photographs and image sequences. In *Toward Category-Level Object Recognition*, pages 105–126, 2006.
- [39] K. Sato, K. Ikeuchi, and T. Kanade. Model-based recognition of specular objects using sensor models. In *Proc. IEEE Workshop on Directions in Automated CAD-Based Vision*, pages 2–10, 1991.
- [40] K. Sato, K. Ikeuchi, and T. Kanade. Model based recognition of specular objects using sensor models. In *Workshop on Directions in Automated CAD-Based Vision, 1991.*, pages 2–10, 1991.
- [41] S. Savarese, M. Chen, and P. Perona. Local shape from mirror reflections. *International Journal of Computer Vision*, 64(1):31–67, 2005.
- [42] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151–172, 2000.
- [43] H. Schultz. Retrieving shape information from multiple images of a specular surface. *IEEE Trans. on PAMI*, 16(2):195–201, 1994.
- [44] S. Shirdhonkar and D. W. Jacobs. Non-negative lighting and specular object recognition. In *ICCV*, pages 1323–1330, 2005.
- [45] D.G. Sim, O.K. Kwon, and R.H. Park. Object matching algorithms using robust Hausdorff distance measures. *IEEE Trans. on Image Processing*, 8(3):425–429, 2002.

- [46] T. Suk and J. Flusser. Graph method for generating affine moment invariants. In *ICPR*, volume 2, 2004.
- [47] R. Swaminathan, S.B. Kang, R. Szeliski, A. Criminisi, and S.K. Nayar. On the Motion and Appearance of Specularities in Image Sequences. In *European Conference on Computer Vision (ECCV)*, volume I, pages 508–523, 2002.
- [48] K.K. Thornber and D.W. Jacobs. Broadened, specular reflection and linear subspaces. Technical Report 2001-033, NEC, 2001.
- [49] J. Wang and K. J. Dana. A novel approach for texture shape recovery. In *ICCV*, page 1374, 2003.
- [50] L.B. Wolff. Using polarization to separate reflection components. In *CVPR*, pages 363–369, 1989.

Aaron Netz received the B.A. degree in 2001 from the faculty of Computer Science at the Technion, Israel. From 2001 to 2007 he served in the Israeli Air Force as a software engineer. He expects to receive his M.Sc in Computer Science in 2012 from the University of Haifa, Israel. Since 2010, he is employed as a software engineer at Slant Six Games, in Vancouver, BC, Canada.

Margarita Osadchy received the PhD degree with honors in computer science in 2002 from the University of Haifa, Israel. From 2001 to 2004, she was a visiting research scientist at the NEC Research Institute. During 2004-2005 she was a postdoctoral fellow in the Department of Computer Science at the Technion - Israel Institute of Technology. In 2005, she joined the Department of Computer Science at the University of Haifa, where she is an assistant professor. Dr. Osadchy's research has focused on computer vision and machine learning especially in the areas of object and event recognition.