Using Specular Highlights as Pose Invariant Features for 2D-3D Pose Estimation

Aaron Netz and Margarita Osadchy
Computer Science,University of Haifa
Carmel, Haifa 31905, Israel
aaron@gmail.com, rita@cs.haifa.ac.il

Abstract

We address the problem of 2D-3D pose estimation in difficult viewing conditions, such as low illumination, cluttered background, and large highlights and shadows that appear on the object of interest. In such challenging conditions conventional features used for establishing correspondence 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. These ideas are incorporated in an efficient method for pose estimation from a monocular image of an object using only highlights produced by the object as its input. The proposed method uses no knowledge of lighting direction and no calibration object for estimating the lighting in the scene. The evaluation of the method shows good accuracy on numerous synthetic images and good robustness on real images of complex, shiny objects, with shadows and difficult backgrounds[1].

1. Introduction

The focus of this paper is 2D-3D pose estimation of shiny objects. We assume that we have a 3D model of the object and the task is to find the pose of the object relative to a calibrated camera from a single monocular image. Determining the pose means finding the 6 parameters of the 3D translation and rotation, which align the projection of the model with the input image.

Much work has been done on this topic for Lambertian objects with prominent texture or shape features under simple lighting conditions, in which all parts of the object are well illuminated (e.g, [16, 9, 20, 17, 31]). The assumptions used in these methods do not hold for an image of a smooth, glossy, textureless object with highlights and shadows, which is placed against a cluttered scene. Figure 1 shows examples of such images. In this work we make use of such challenging conditions as specular highlights produced by a known object, to extract information that can assist in pose estimation when other cues are unreliable. Specularities have several advantages over the conventional features used for pose estimation. Highlights are easy to detect even by simply thresholding the image. 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.

Previous methods that used specular cues for pose estimation required additional information about the scene,

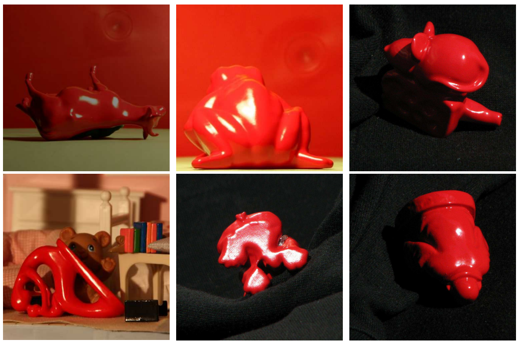

Figure 1. Examples of typical inputs to the proposed pose estimation algorithm, which are very challenging for the existing methods.

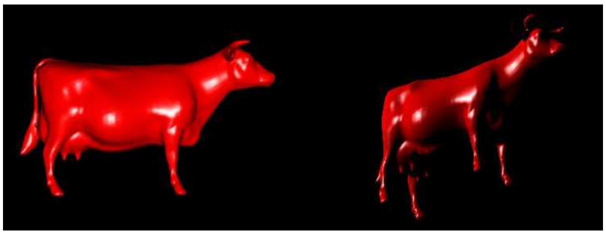

Figure 2. Left – the frontal view of the object with the frontal light(that we render), right – unknown view of the object with unknown light (that we get as an input). In both the highlights are produced by the same surface patches and thus highlights undergo affine transformation and could be used for establishing correspondence.

[1] the data base is available at http://www.cs.haifa.ac.il/ rita/specObj/main.html

such as environmental mapping obtained by a mirror ball [8], known motion [8], polarization filters [3], and several images of the scene with different camera settings [3]. If such additional cues are unavailable, these methods cannot be used.

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 (Figure 2). Based on this observation, we suggest to use specular highlights produced by a known, smooth, glossy object to establish correspondences between its image and the 3D model without any knowledge of the scene except for the assumption of the dominant light source (the direction of the light is unknown). We use these correspondences to compute a pose, and we verify it by measuring the similarity between the specular highlights extracted from the input image and the specularities predicted for the hypothesized pose using a simple model of highlight formation proposed in [26]. Our method allows to estimate pose from a monocular image of an object using solely specular features if the image contains at least three highlights. If the image contains only one or two highlights, our approach can easily incorporate correspondences obtained from other more conventional features (contours, lines, etc.) for pose computation, while the verification phase remains unchanged.

Our approach is advantageous over the previous work on pose estimation of specular objects because it requires only a single monocular image of the object and it can work with shapes that are much more complex than those used in previous works ([8, 3]). Also our method doesn't require knowledge of the lighting direction in the scene, or any calibration object (or procedure) for estimating it. The proposed algorithm is based on correspondences, which is much more efficient than a brute force search done by previous methods (e.g.,[8]).

The main limitation of our approach is the assumption of the dominant light source. In practice, however, it is a reasonable assumption for outdoor scenes. For indoor scenes, which are illuminated by many sources or extended lights, the object is well illuminated and even if it has highlights, existing methods (e.g., contour based methods) could work fairly well. Images taken with a directional light are poorly illuminated and have large shadows, which present a severe difficulty for the existing methods.

The experiments presented in this paper are performed on synthetic and real objects. We constructed a data base 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. The experiments (see Section 5) show good performance considering that our method uses very little information about the scene and only few percents of the input image – the highlights.

1.1. Related Work

Most of the work on specular objects has been concerned with surface reconstruction [5, 6, 33, 25, 36]. Norman et. al., [24] 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 [32, 15, 26]. Recently, several methods have been proposed for detecting specular surfaces in images [21, 12]. 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 [19], which showed very impressive results. Only very recently specular highlights have been used in pose estimation [8].

The literature concerned with 3D pose estimation is extensive. One of the aspects that allows to distinguish between various method for pose estimation is the type of local image features they use to establish correspondences, such as points, lines or segments (e.g.,[28, 2]), curved shapes or their segments (e.g,[7]), and complete contours (e.g.,[31]). A more recent development in pose estimation uses regions in a global variational framework (e.g.,[30, 11]). Fusion of several information channels was suggested for pose estimation in [23, 3]. Using depth information of the scene could greatly assist in solving the pose problem, thus quite many methods use range images as an input (e.g., [4], [14]).

Using specular cues for pose estimation was considered in [3, 8]. The method presented in [3] 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 work that is most relavant to ours [8] renders 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 (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 [8] are done on simple objects with complex illumination, which strongly constrains the appearance of highlights. The same work [8] 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.

2. Basic Approach

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

Let P define a planar, mirror-like patch in a 3D space with normal $\vec{N}$. Assume that the patch is illuminated by a single, distant, compact light source and the distance to the camera is large enough to assume weak perspective projection.

Claim 1: For each combination of viewing direction $\vec{V}$ and light direction $\vec{L}$ that produce specular reflection on P, there exists another viewing direction $\vec{V'}$ and light direction $\vec{L'}$ such that $\vec{L'} = \vec{V'}$, for which P remains specular ($\vec{N} = \vec{L'} = \vec{V'}$). The proof follows immediately from the standard models of specular reflection [10, 37, 27].

Claim 2: Let p be an image of P corresponding to illumination direction $\vec{L}$ and viewing direction $\vec{V}$, for which $\vec{N}$ is a bisector of the angle between $\vec{L}$ and $\vec{V}$. Let p' be an image of P corresponding to illumination direction $\vec{L'}$ and viewing direction $\vec{V'}$, such that $\vec{L'} = \vec{V'} = \vec{N}$. Then p and p' are related by an affine transformation. (Under the assumption of weak perspective projection the proof is trivial.)

Now consider a 3D smooth object with a specular but non-mirror reflectance, which is illuminated by a single, distant, compact light source. According to most models of specular reflection [27, 10, 37, 22], a ray of light reaching a shiny surface is reflected as a narrow beam of rays. The brightest direction of reflection $\vec{R}$ will lay at the plane formed by the original ray of light $\vec{L}$ and the normal at the point of impact $\vec{N}$ and will form an angle defined by $\vec{L} \cdot \vec{N}$ relative to $\vec{N}$, on that plane. The intensity of the reflected rays decays as they deviate from $\vec{R}$. The rate of decay is determined by the shininess of the surface. According to the well known Phong model [27], the intensity of a reflected ray in direction $\vec{V}$ is proportional to $(\vec{V} \cdot \vec{R})^{\alpha}$, where α denotes the shininess of the surface. We further simplify the model by assigning only binary intensity values: the intensity at a point with normal $\vec{N'}$ is set to 1 if $\vec{N'} \cdot \vec{N} > t$ and is set to 0 otherwise. The threshold t depends on the shininess of the material and is clearly related to α. For high values of t the highlighted part of the surface can be approximated by a planar patch with orientation corresponding to the normal $\vec{N}$. Under this assumption, we can extend Claims 1 and 2 to a 3D surface:

Claim 3: Under the assumption of the dominant light source, a 3D patch of a smooth, specular, non-mirror object that appears highlighted in an image with unknown illumination and pose will also appear highlighted in an image for which $\vec{L} = \vec{V} = \vec{N}$ ($\vec{L}$ and $\vec{V}$ are the light and the viewing directions and $\vec{N}$ is the central normal), and these highlights are related by an approximately affine transformation.

Corollary: Knowing the specular properties of the object allows us to render highlights for every surface normal such that $\vec{L} = \vec{V} = \vec{N}$. Given this set, we can relate a highlight in an image with unknown pose and illumination direction to the surface patch that produced it by applying an affine invariant matching between the real highlight and the rendered ones.

As we will soon elaborate, the matching between a highlight from an unknown view to the patch that produced it, can be done efficiently. We can then assume that the centroid of the highlight in the image and the centroid of the corresponding 3D patch is approximately the same point. Consequently, we 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. Since we only use a single point from a highlight for establishing the correspondence – the centroid, we need at least three highlights in the image for finding the pose solely from specularities. However, if other cues are available (prominent shape or texture features), we can easily integrate the correspondences obtained from different sources to find the candidate pose. We show later that the verification of the pose uses only specularities and it has no limitation on the number of highlights (even a single highlight could suffice).

An efficient way of matching between a highlight in an image with unknown lighting and pose and a 3D patch that produced it, includes the following steps. During the offline stage we first render highlights as viewed from each point on the viewing sphere (according to some tessellation) for the special case in which the lighting direction coincides with the viewing direction. We then compute the affine invariant descriptors of the rendered highlights for every view (affine invariants are computed for each highlight separately since the highlights that have different depth do not lie in the same plane) and store them indexed by the viewing direction. Given an input image, we compute the affine invariant descriptors of the highlights in that image and search for a candidate view by matching the invariants in the given image to the pre-computed invariants of the rendered highlights (according to Claim 3).

Matching all highlights that appear in a view as a set, as opposed to matching each highlight individually, has its positive and negative sides. The positive side is that matching a set of highlights has lower chance to false matches compared to matching an individual highlight; and it's computationally more efficient. The negative side is that a rendered view and an input image could have different number of highlights due to self occlusions. To solve this problem, our matching procedure allows for unmatched highlights (See Section 4).

We choose a portion of candidate views that best match the invariants computed from the input image. For each candidate view we extract the correspondences and compute a hypothesized pose. We choose the pose with highest verification score (see step 5 in Section 4) among the candidate poses. The exact number of candidate views needed for correct pose estimation depends on the object. If the object has very complex shape and most of its local parts differ one from 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, our method remains efficient. First, because we match sets of highlights instead of individual highlights, which decreases the number of false matches. Second, for every candidate view we apply polynomial matching (See Section 4) in order to establish correspondence between the highlights in this view to the highlights in the image, and compute a *single* hypothesized pose for the view (which is much more efficient than computing pose for all possible correspondences of the highlights).

2.1. Pose from correspondences

Since specular highlights are sparse, we need a method that works with a minimal number of correspondences. We therefore employ the method from [1], since it is computationally efficient and needs only 3 correspondences. In practice, there are not many cases in which there are more than 3 significant highlights. In such cases, we run this algorithm on all possible triplets of correspondences, and take the correspondence which gives the lowest error on the rest of the points.

2.2. Affine invariants

Our method uses affine invariants for finding correspondence between the highlights in the image and the set of 3D points that produced the highlight. Much work has been done on affine invariants and their use in computer vision. A survey of this work is beyond the scope of the paper. We chose Affine Moment Invariants [13, 35] due to computational efficiency and low storage requirements. Given a binary image of a highlight, cropped from the image of the object, we construct the 17 independent invariants up to weight 8 as polynomials in the central moments of the image. We then combine them into a single vector and use it as an affine invariant descriptor of the highlight.

3. Pose Estimation Algorithm

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 a preprocessing phase.

3.1. Offline stage

Given a 3D model of an object, we 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 $\vec{N}_i$ we perform the following steps:
Step 1. We set $\vec{V} = \vec{L} = \vec{N}_i$ ($\vec{V}$ is the viewing and $\vec{L}$ is lighting directions), meaning that $\vec{N}_i$ is the center of the specular beam. We render a binary image B_i of the object from the viewing direction $\vec{V}$ according to the model introduced in Section 2: the intensity of a pixel is set to one if the dot product between its normal and $\vec{N}_i$ is larger than a predefined threshold t , otherwise the intensity is set to zero.
Step 2. We locate significant highlights in B_i by finding the connected components and removing very small ones, since their affine invariants are unstable due to discretization.
Step 3. We compute an affine invariant descriptor [13, 35] for every significant highlight in B_i, along with the 3D centroid of the surface points that produced it. The descriptors and 3D centroids are stored for each normal $\vec{N}_i$.

4. Online stage

During the online stage we are presented with an image denoted by I.
Step 1. We extract specular highlights in I. We do it by first applying a high threshold on I, and then a low threshold but selecting only the highlights which intersect with those that passed the high threshold (Figure 3). This method worked well in all our tests because it extracted the entire highlight and not only the saturated pixels[2]. It can be replaced, however, with any other method for highlight segmentation. Let B_I denote the binary image of the highlights. Next we determine the significant highlights in B_I as was explained in Step 2 in Section 3.1.
Step 2. For each significant highlight in B_I we calculate the affine invariant descriptor [13, 35] and the 2D centroid. 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.
Step 3. In this step we find correspondences between the highlights in I and the 3D model. Specifically, we find a set of candidate views that best match the highlights in I.

[2] If the dynamic range of the input image is low and there are parts of the background that have the same intensities as the highlights, we could apply the same heuristic but using two images of the scene with short and long shutter speeds.

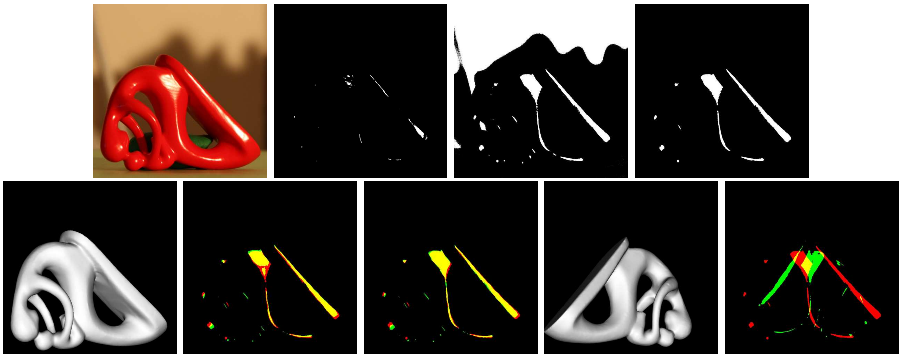

Figure 3. Step 1 and 5 of the online stage. Top row, left to right: a given image with unknown pose and lighting, high threshold binary image, low threshold binary image, highlights from low threshold which intersect with those that passed the high threshold; Bottom row, left to write: correct hypothesized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

To this end for each viewing direction according to the tessellation, we construct a full bi-partite graph in which one side corresponds to the highlights in I and the other to the highlights stored for that view. The weights on the edges are the Euclidian distances between the descriptors. We use Hungarian algorithm [18] to find the best maximum match. Matching also relates between the 2D centroids of the highlights in I and the 3D centroids of the surface points, which provides the 2D-3D correspondence needed for the pose estimation algorithm. Next we choose up to K views, with matching score higher than a predefined threshold, as candidates for the correct correspondence (the matching score is computed as minus average distance between the descriptors of the matched highlights). The threshold and K are chosen empirically. The number of significant highlights in every view is small, which makes the matching very fast.

Step 4. For each candidate correspondence, we find the pose as shown in section 2.1.

Step 5. We run a verification process on each pose, obtained by Step 4. The hypothesized pose allows us to match image pixels to corresponding surface normals on the model. We map each pixel in B_I to a point on a Gaussian sphere having the same surface normal, while giving different colors to specular and non-specular pixels. According to the model introduced in [26], if the pose is correct, the normals corresponding to the specular pixels must form a cap on a Gaussian sphere and the size of the cap is determined by the material properties of the object. Since these are known, we could adjust the coloring on the sphere such that the specular normals form a cap of the correct size. The size of the cap can be controlled by 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 $\vec{v'}$, for which the set of specular normals $\{\vec{v} \mid \vec{v} \cdot \vec{v'} > t\}$ is the largest. We choose $\vec{v'}$ to be the center of the cap and set all normals $\vec{v}$ satisfying $\vec{v} \cdot \vec{v'} > t$ to be specular. The updated coloring is then mapped back to the image plane and compared with B_I . This process relies on the fact that if the hypothesized pose is correct, the updated highlights will be similar to the original, but if the hypothesized pose is wrong the updated highlights will be inconsistent with the original (Figure 3). The overlap measure used in [26] is not robust to small shifts, caused by the errors in pose. Thus we applied a robust variant of Hausdorff measure [34] to compare the binary images: $H(B_I, B'_I) = h(B_I, B'_I) + h(B'_I, B_I)$, where B'_I is the binary image of highlights mapped back from the Gaussian sphere and

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

where $|A|$ is the number of non-zero pixels in A and α is a constant, depending on the size of the image (choosen empirically).

Step 6. (Optional) We found that running an optimization of the verification function, with the hypothesized pose as a starting point is helpful for refining the pose. We take S hypothesized poses with the best verification score and run a standard routine for constrained non-linear optimization [29] using these poses as starting points. The pose that produces the best score (after optimization) is the output of the algorithm[3].

5. Experiments

We start by providing the implementation details and then show the results on real and synthetic objects.

5.1. Implementation Details

In all our experiments we used 3D models available on the Web [4]. The 3D models were centered, bound to the unit

[3] due to running time constraints we do not run the optimization for every hypothesized pose; in our experiments we set $S = 3$

[4] See http://shapes.aim-at-shape.net/

				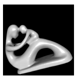	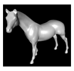				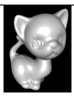		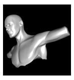
Av.Success Rate Synth. (%)	71	80	90	70	85	85	70	70	100	85	90
Av. Transl. Error Synth.	0.028	0.013	0.015	0.010	0.012	0.017	0.026	0.016	0.012	0.020	0.011
Av. Rotation Error (deg.) Synth.	4.72	3.09	2.96	4.35	2.82	2.85	4.64	4.35	2.30	4.09	1.64
Av.Success Rate Real (%)	69	78	87	70	69	69	54	55	80	77	54
Av. Transl. Error Real	0.077	0.073	0.039	0.051	0.032	0.036	0.067	0.104	0.078	0.060	0.037
Av. Rotation Error (deg.) Real	11.85	11.028	8.39	8.55	5.40	3.87	8.67	9.81	9.40	6.15	9.57

Table 1. Pose estimation results. The top three rows correspond to the synthetic set (total 220 poses). The bottom three rows correspond to real outdoor set (total 156 poses). The units of the translation error are relative to the object size.

sphere in size, and remeshed to have between $50,000$ and $100,000$ faces. Processing of the models was done using MeshLab [5].

For the offline stage of the algorithm, the rendering of highlights was done at a resolution of 1024×1024. Both synthetic and real input images were of the same resolution. For the verification step (see step 5 in Section 4), rendering of mapped-back highlights was done at a resolution of 256×256.

The algorithm was implemented mostly in MATLAB and partly in Java for the OpenGL renderings. The average (online) running time of the algorithm was around 60 seconds, which could be significantly improved by more efficient implementation and also by parallelization, which is possible during most stages of the algorithm.

Determining t: In order to find t for a given object we use the 3D model of the object and its image in a known pose. We segment the highlights by applying the method used in the online stage (step 1, Section 4). We then map the pixels within the highlights to the points on the Gaussian sphere having the same surface normal. According to the model from [26] the specular points of the sphere must form a cap, which we find using the algorithm from [26]. We set 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.

5.2. Synthetic images

We have tested our algorithm on 11 complex objects, with different levels of shininess. Table 1 shows the images of the objects with their selected levels of shininess. For each object, we have generated 20 random poses, restricting them to have at least three highlights.

The output of the algorithm was evaluated separately for translation error and rotation error. Denote the true translation vector as τ 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 $\tilde{R}$. A successful output pose was considered a pose whose translation error is less than 0.08 (in units, relative to the object size) and rotation error is less than 20 degrees (which is roughly equivalent to 10 degrees error for all 3 Euler angles). Table 1 shows the average success rates for each of the 11 objects and the average translation and rotation errors for successful output poses.

5.3. Real Images

We used 3D models from the synthetic experiment to create real objects using 3D printing technology that allows to produce objects from a CAD model with relatively high accuracy. These objects were painted with a glossy paint, which produces specular effects. We colored all 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. We constructed a data set from 225 real images, divided into two subsets: outdoor and indoor. The number of poses for each object corresponds to the number of images of that object. The outdoor set contains 157 images of all 11 objects against black background. The variation in light direction in these images are due to sun movement, and thus is not very large. The indoor subset contains 68 images of 5 objects: cow, mouse, fertility, gargoyle, and frog. The photos were taken against both plain and cluttered

[5] See http://meshlab.sourceforge.net/

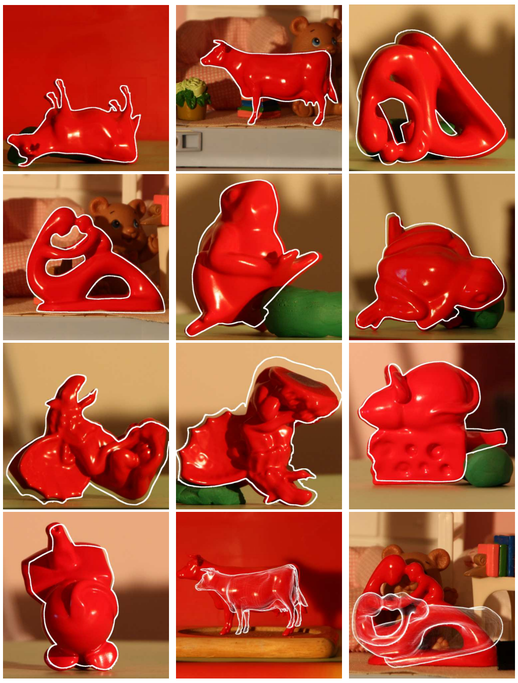

Figure 4. Examples of the pose estimation results on real indoor images. The white contour corresponds to the occluding contour of the object in the estimated pose. The two images in the bottom right corner show the failure cases.

		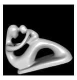			
Success(%)	81	63	55	45	57

Table 2. Average success rate of the pose estimation on the indoor real subset.

backgrounds and include large variation in illumination direction.

We manually labeled 2D-3D correspondences for the outdoor set, and used them to compute poses. Since the objects are smooth and textureless our manual correspondences are not exact, and thus the pose computed using these correspondences cannot be considered as the ground truth. However, they are accurate enough for evaluating the automatic algorithm for pose estimation. Table 1 reports the results in the same format as in Section 5.2 using the manual poses as true poses. We do not have manual labeling for the indoor set. Thus we classified the output of the proposed method as success of failure by visually comparing an image, rendered with the computed pose to the corresponding real images. Table 2 shows the success rates for the indoor set. The average recognition rate over all real images is 67.5%. Note that according to our definition of success (See Section 5.2) the probability of a randomly chosen pose to be considered correct is much less than 1%.

6. Conclusions

In this work we addressed a challenging task of pose estimation 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 lights source, specular highlights could be used as a pose invariant features. We developed a pose estimation algorithm that relies solely on highlights and doesn't require the knowledge of lighting. The proposed method showed good results in evaluation that included synthetic and real images.

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 results 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. We plan to explore this and other optimizations in future work in order to use the proposed method for recognition. We also plan to extend the proposed approach to more general illumination and integrate other cues for correspondence such as prominent texture and shape features.

References

[1] Alter. *3d pose from 3 corresponding points under weak-perspective projection*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1992.

[2] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[3] B. Barrois and C. Wohler. 3d pose estimation based on multiple monocular cues. In *BenCOS07*, pages 1–8, 2007.

[4] P. J. Besl and H. D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] A. Blake and G. Brelstaff. Geometry from specularities. In *Second International Conference on Computer Vision*, pages 394–403, 1988.

[6] 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.

[7] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[8] J. Y. Chang, R. Raskar, and A. K. Agrawal. 3d pose estimation and segmentation using specular cues. In *CVPR*, pages 1706–1713, 2009.

[9] J. Chen and G. Stockman. Matching curved 3D object models to 2D images. In *CAD-Based Vision Workshop, 1994., Proceedings of the 1994 Second*, pages 210–218, 1994.

[10] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. In *SIGGRAPH*, pages 307–316, 1981.

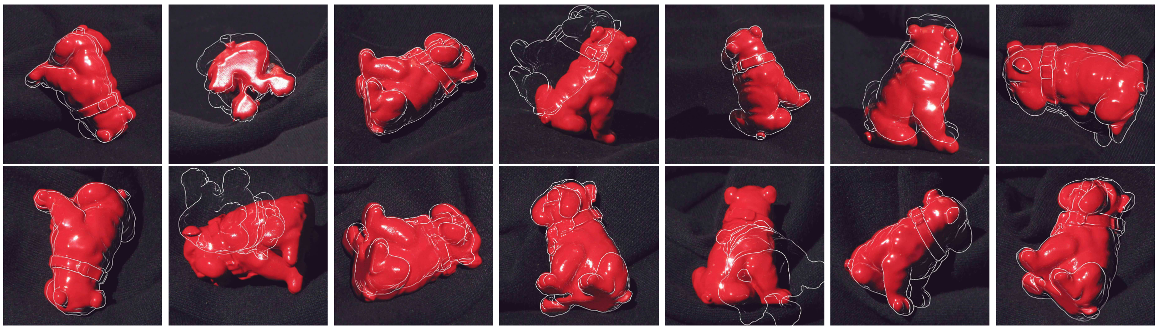

Figure 5. Variation of poses for one of the objects from the outdoor subset. The pose estimations shown by white contours are overlaid with the original images.

[11] S. Dambreville, R. Sandhu, A. J. Yezzi, and A. Tannenbaum. Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior. In *ECCV (2)*, pages 169–182, 2008.

[12] A. DelPozo and S. Savarese. Detecting specular surfaces on natural images. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Volume 2)*, 2007.

[13] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern recognition*, 26(1):167–174, 1993.

[14] M. Germann, M. D. Breitenstein, I. K. Park, and H. Pfister. Automatic pose estimation for range images on the gpu. In *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 81–90, 2007.

[15] K. Gremban and K. Ikeuchi. Planning multiple observations for object recognition. *International Journal of Computer Vision*, 12(2):137–172, 1994.

[16] D. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. ICCV*, pages 102–111, 1987.

[17] D. Jacobs and R. Basri. 3-d to 2-d pose determination with regions. *International Journal of Computer Vision*, 34(2):123–145, 1999.

[18] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[19] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua. 3d pose refinement from reflections. In *CVPR*, 2008.

[20] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[21] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Volume 2)*, pages 973–979, 2005.

[22] S. Nayar, K.Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):611–634, Jul 1991.

[23] Y. Nomura, D. Zhang, Y. Sakaida, and S. Fujii. 3-d object pose estimation base on iterative image matching: Shading and edge data fusion. *International Conference on Pattern Recognition*, 1:513, 1996.

[24] J. Norman, J. Todd, and G. 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.

[25] M. Oren and S. Nayar. A theory of specular surface geometry. *International Journal of Computer Vision*, 24(2):105–124, 1997.

[26] M. Osadchy, D. Jacobs, R. Ramamoorthi, and D. Tucker. Using specularities in comparing 3D models and 2D images. *Computer Vision and Image Understanding*, 111(3):275–294, 2008.

[27] B. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[28] T. Q. Phong, R. Horaud, A. Yassine, and P. D. Tao. Object pose from 2-d to 3-d point and line correspondences. *Int. J. Comput. Vision*, 15(3):225–243, 1995.

[29] M. Powell. A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis, G.A.Watson ed., Lecture Notes in Mathematics*, 630, 1978.

[30] B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *Int. J. Comput. Vision*, 73(3):243–262, 2007.

[31] B. Rosenhahn, C. Perwass, and G. Sommer. Pose estimation of 3d free-form contours. *Int. J. Comput. Vision*, 62(3):267–289, 2005.

[32] K. Sato, K. Ikeuchi, and T. Kanade. Model based recognition of specular objects using sensor models. In *Automated CAD-Based Vision, 1991., Workshop on Directions in*, pages 2–10, 1991.

[33] H. Schultz. Retrieving shape information from multiple images of a specular surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):195–201, 1994.

[34] D. Sim, O. Kwon, and R. Park. Object matching algorithms using robust Hausdorff distance measures. *Image Processing, IEEE Transactions on*, 8(3):425–429, 2002.

[35] T. Suk and J. Flusser. Graph method for generating affine moment invariants. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, 2004.

[36] J. Wang and K. J. Dana. A novel approach for texture shape recovery. In *Proceedings of the Ninth International Conference on Computer Vision*, page 1374, 2003.

[37] G. J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH*, 26(2):265–272, 1992.