Automatic Estimation of the Projected Light Source Direction

Peter Nillius Jan-Olof Eklundh

Computational Vision & Active Perception Laboratory (CVAP)
Department of Numerical Analysis and Computing Science
Royal Institute of Technology (KTH), S-100 44 Stockholm, Sweden

Abstract

We present a fully automatic algorithm for estimating the projected light source direction from a single image. The requirement is that there exists a segment of an occluding contour of an object with locally Lambertian surface reflectance in the image. The algorithm consists of three stages. First a heuristic algorithm picks out potential occluding contours using color and edge information. Secondly, for each contour the light source direction is estimated using a shading model. In the final stage the results from the estimations are fused together in a Bayesian network to arrive at the most likely light source direction. The probabilistic model takes into account that the contours from the first stage might not be occluding contours. Using the same framework the contours are also classified as occluding or not. Experiments test the second stage, estimating the light source direction from an occluding contour, as well as the full algorithm.

1 Introduction

The occluding contour is where an object occludes itself like the earth at the horizon. At the occluding contour the shape of the object can easily be determined because the surface normal is perpendicular to the viewing vector and can be determined by the image edge direction [3].

Many existing algorithms for estimating the light source direction use occluding contours. In [7] it is a requirement that the image is of a convex object bounded by an occluding contour. [10] uses the same occluding contour assumption to match default shapes to the image and then estimate the light source direction. Both methods require Lambertian surfaces and a segmented image of the object. [12] also exploit the occluding boundary and show that it puts strong constraints on the light source direction.

Recent algorithms use known geometry to derive the illumination direction. E.g. [8, 9] derive the illumination distribution by studying shadows around an object of known geometry. In [13] multiple light sources are extracted from a sphere of known size.

Previous work depends on segmented images or known geometry. In this article we will not use these assumptions, but only determine the projected light source direction. We will present a fully automatic algorithm for recovering this projected direction. First the estimation of the light source direction using the shading near the occluding contour is investigated. Also, the noise distribution is derived. Secondly an algorithm for picking out potential occluding contours using edge and color information is presented. The contours produced by this algorithm are then used in a Bayesian probabilistic framework to estimate the most likely light source direction. Simultaneously, the contours are classified as occluding or not.

2 Shading at the Occluding Contour

First we will look at the case of estimating the light source direction from an occluding contour.

Given that the illumination is a single point light source, the model for the image intensity I at a point on a Lambertian surface is

$$I = k(\vec{n} \bullet \vec{l}) + a \qquad (1)$$

where $\vec{n} = (n_x, n_y, n_z)^T$ is the surface normal at the point, $\vec{l} = (l_x, l_y, l_z)^T$ is the direction to the light source, k is a parameter containing both the surface albedo and the strength of the light source, a is a term representing the contribution of the ambient illumination.

On the occluding contour n_z is equal to zero. This will eliminate l_z from the equation, which is why we will not be able to estimate l_z. Due to the bas-relief ambiguity [1] the z-component of the light source direction cannot be estimated using only this model when the surface albedo and light source strength is unknown. Now, the image intensity on the occluding contour will be

$$I = k(\vec{n} \bullet \vec{l}) + a = k(n_x l_x + n_y l_y) + a. \quad (2)$$

From the image we can measure $\vec{n}$ and I. $\vec{l}$, a and k are unknown, but are assumed to be constant for each local computation. Since we only are interested in the direction to the light source scaling of the $\vec{l}$ vector doesn't matter. Therefore let $x = kl_x$ and $y = kl_y$. Now we have

$$I = n_x x + n_y y + a = \begin{pmatrix} n_x & n_y & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ a \end{pmatrix} \quad (3)$$

Equation (3) has three unknowns x, y and a and as many equations as the number of points on the occluding contour. Using e.g. least squares we can estimate the light source direction. Let $\mathbf{N}$ be a matrix containing the n_x and n_y for the points along the contour and $\mathbf{I}$ a vector containing the intensity values for the same points. Then the least-squares estimate $\hat{L} = (\hat{x}, \hat{y}, \hat{a})^T$ is calculated by

$$\hat{L} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{I}. \quad (4)$$

In $\hat{L}$ the estimated angle towards the light source is found as the direction of the vector $\hat{l} = (\hat{x}, \hat{y})^T$, i.e.

$$\hat{\phi} = \arg(\hat{l}). \quad (5)$$

2.1 Measuring the intensity at the contour

To be able to estimate the light source direction in this way we need to measure the intensity at the occluding contour. This is of course impossible, the intensity cannot be measured at the contour. Also, the Lambertian model becomes inaccurate very close to the contour [6, 11]. To overcome this we will look at the intensities some distance away from the contour and extrapolate to get the intensity at the contour.

The extrapolation is done by modeling the image intensities along a line, the u-axis in Figure 1a, perpendicular to the edge. By measuring the intensities along the line and using the model, the intensity at the edge can be estimated by extrapolation.

To model the intensities, a model of the shape along the line is needed. We have chosen an ellipse to model the cross-section of the object, see Figure 1b. Because of the bas-relief ambiguity the z-axis can be scaled arbitrarily and the shape model can be simplified as a circle, with radius R, without any loss of generality in this case. We have

$$(u - R)^2 + z^2 = R^2 \quad (6)$$

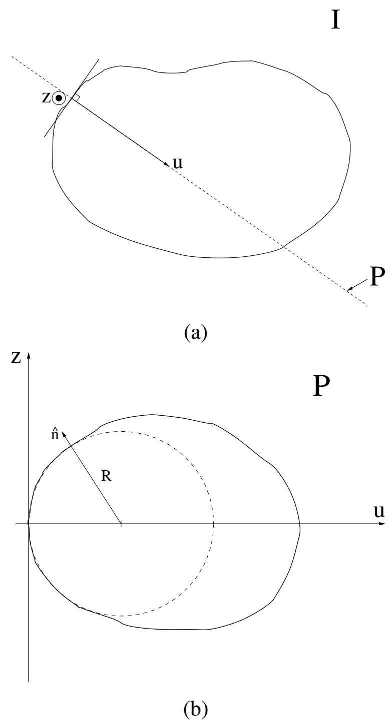

Figure 1. a) An image I of an object. To be able to do extrapolation a model of the shape of the object along the u-axis is needed. This is done by modeling the cross-section b) of the object in the plane P with an ellipse.

from which we can express z as function of u and the normal of the surface as a function of u and z.

$$\begin{aligned} z &= \sqrt{2uR - u^2} && (7) \\ N(u) &= \frac{1}{R}\begin{pmatrix} u - R \\ z \\ 0 \end{pmatrix} && (8) \end{aligned}$$

Inserting the expression for the normal into the intensity equation (1) gives us the intensity as a function of u.

$$I(u) = \frac{k}{R}\left(l_u(u - R) + l_z\sqrt{2uR - u^2}\right) + a \quad (9)$$

This model is too elaborate to use for extrapolation. The parameters in fact include the direction towards the light source, which is what we want to estimate. A power series expansion of the model can however be derived. With the power series we can select the models level of detail by including more or less terms from the series.

The power series of the model (9) is of the form:

$$\begin{aligned} I(u) = {} & c_0 + c_1 u^{1/2} + c_2 u + c_3 u^{3/2} + c_4 u^{5/2} \\ & + c_5 u^{7/2} + O(u^{9/2}). \quad (10) \end{aligned}$$

The extrapolation is done by fitting the power series polynomial to the measured intensities using least-squares estimation and then calculating the intensity at $u = 0$ using the estimated polynomial, i.e. $I(0) = c_0$.

2.2 The Probability Distribution of the Estimates

The estimated vector $\hat{L}$ is a linear combination of image intensities. The extrapolation operation and the estimation of the direction are both linear operations. In general each estimate will be a linear combination of hundreds of image intensities [1], which means that the estimated vector is approximately normally distributed. Since least-squares is a non-biased estimator, the mean will be $(x, y, a)^T$.

$$\hat{L} \sim N\left(\begin{pmatrix} x \\ y \\ a \end{pmatrix}, \Sigma\right) \tag{11}$$

The covariance matrix Σ of a least-squares estimate is

$$Cov(\hat{L}) = (\mathbf{N}^T\mathbf{N})^{-1}\sigma^2. \tag{12}$$

The noise variance σ^2 can be estimated with

$$\hat{\sigma^2} = \frac{1}{n-1}\sum_{i=0}^{n}(I_i - (n_x\hat{x} + n_y\hat{l_u} + \hat{a}))^2. \tag{13}$$

3 An Automatic Algorithm

The automatic algorithm has three stages. First a heuristic algorithm picks out candidate occluding contours using color and edge information. Secondly, for each of the contours, the light source direction is estimated according to the method presented in the previous section. In the final stage the estimates are fused in a Bayesian network to arrive at the most likely light source direction.

3.1 Finding Potential Occluding Contours

The goal of this stage is to, in a heuristic way, pick out contours of which as many as possible are occluding contours. The simple rules used here will naturally not be able to pick out occluding contours perfectly, but they will provide the later stages with good enough candidates.

The estimation of the light source direction described in Section 2 only works on occluding contours of uni-colored objects. The algorithm should pick out edge chains that have an uni-colored area perpendicular to the chain direction. Also, for the extrapolation not to be disturbed there should be no edges in that same area. The surface of the object needs to be smoothly curved so a sharp turn of the contour is a strong indication that the contour is not from a useful occluding boundary.

The algorithm works in the following way. First the edges are extracted, using the Canny edge detector, and linked into chains. By following the chains the potential occluding contours are picked out by grouping together consecutive edges if

- The area next to the edge is uni-colored
- The color is the same as previous edges in the chain
- The area next to the edge contains no other edges
- The chains does not make a sharp turn

This is done on both sides of the edges. Very straight contours contain no information for our algorithm and are sifted out in a post-processing stage.

The test whether a surface is uni-colored is done in a similar way as in [5]. For each piece of a contour segment the pixels next to it are analyzed by a singular value decomposition of the pixel cluster in RGB-space. If the two eigenvectors with the lowest eigenvalues are small enough then the pixels follow the model $\vec{c_i} = k_i\vec{c}$ and are assumed to be on a uni-colored object.

The algorithm is very simple and does an acceptable job for our purposes at this stage, but there are some problems. The main problem is that contours get split up due to edges in the background. What happens is that, in the edge-linking process, the object contour gets linked together with an edge in the background instead of continuing along the contour. This could in the future be solved by using a more clever grouping algorithm incorporating e.g. good continuation. See Figure 10 a and d for examples.

For each of the contours picked out by the algorithm, the light source direction is estimated as described in Section 2.

3.2 Fusing the Estimates

At this stage we have a set of estimates and variances from n edge chains that might or might not be from occluding contours. To separate the correct contours from the incorrect ones it is necessary that the correct contours in general have a smaller variance , i.e. fit the model better, than the incorrect ones. Also, if we have several estimates pointing towards the same light source direction one can draw the conclusion that this is the correct direction even though their variances aren't substantially smaller than those of the the other contours. The final stage of the algorithm fuses the estimates incorporating these conditions.

[1] In the experiments the extrapolation uses 7 pixels per edge point and the average contours length is about 60 which means that each estimate is a linear combination of over 400 pixel values.

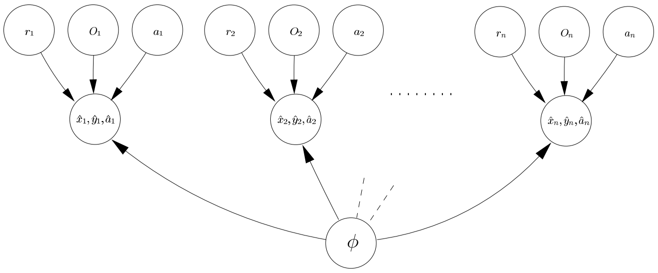

Figure 2. Bayesian network model

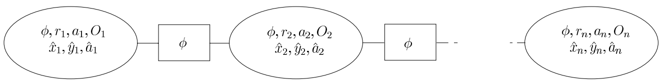

Figure 3. Junction tree

3.2.1 The Probabilistic Model

An estimated vector $\hat{L}_i = (\hat{x}_i, \hat{y}_i, \hat{a}_i)^T$ from contour i depends on the true values $L_i = (x_i, y_i, a_i)^T$ and whether or not the contour is an occluding contour or not, captured in the discrete variable O_i.

Because only the direction of the light source vector is the same for all the contours (the magnitude depends on surface albedo which may be different for different objects), the variables x_i and y_i are represented by their magnitude r_i and direction ϕ. Figure 2 shows the causal dependencies between the different variables.

3.2.2 The Inference Process

What we would like to find is the set of values to the variables that maximizes the probability. Especially we would like to find the ϕ in that set. I.e. if $U = \{\phi, r_1, \ldots, r_n, a_1, \ldots, a_n, O_1, \ldots, O_n, \hat{L}_1, \ldots, \hat{L}_n\}$ we would like to estimate ϕ with

$$\hat{\phi} = \underset{\phi}{\operatorname{argmax}} \max_{U \setminus \phi} P(U) \tag{14}$$

Using message propagation this $\hat{\phi}$ can be found with only local computations [2, 4].

By moralizing the graph, i.e. connecting parents with a common observed child, cliques can be identified. The cliques are the smallest sets of variables on which local computations can be done. Figure 3 shows the resulting junction tree with cliques $C_i = \{\phi, a_i, r_i, O_i, \hat{x}_i, \hat{y}_i, \hat{a}_i\}, i = 1, \ldots, n$ one for each contour and sepsets $S_j = \{\phi\}, j = 1, \ldots, n-1$ containing the common variables of the neighboring cliques.

The initial distribution for each clique will be

$$\begin{aligned} P_0(C_i) &= P_0(r_i, \phi, a_i, O_i, \hat{x}_i, \hat{y}_i, \hat{a}_i) = \\ &= P(r_i)P(\phi)P(a_i)P(O_i)P(\hat{x}_i, \hat{y}_i, \hat{a}_i | r_i, \phi, a_i, O_i), \end{aligned} \tag{15}$$

where $P(r_i)$, $P(\phi)$, $P(a_i)$ and $P(O_i)$ are prior distributions. $P(\hat{x}_i, \hat{y}_i, \hat{a}_i | r_i, \phi, a, O_i)$ is the distribution of the estimates. How these will be assigned will be discussed Section 3.2.4.

Because the junction tree here has a simple structure, the clique probability distribution after the message propagation has a closed form solution, namely

$$P(C_i) = P_0(C_i) \prod_{j \neq i} M_j(\phi). \tag{16}$$

where M_j is the max-margin of the initial distribution of clique j, defined as

$$M_j(\phi) = \max_{C_i \setminus \phi} P_0(C_i). \tag{17}$$

To find the most likely light source direction we can select an arbitrary clique and find the ϕ giving the maximum probability.

$$\hat{\phi} = \underset{\phi}{\operatorname{argmax}} \max_{C_i \setminus \phi} P(C_i) \tag{18}$$

3.2.3 Classifying the Contours

From the clique potentials we can also estimate O_i i.e. classify whether the contour is an occluding contour or not. From the clique potential after the message propagation we classify contour i by

$$\hat{O}_i = \underset{O_i}{\operatorname{argmax}} \max_{C_i \setminus O_i} P(C_i). \tag{19}$$

3.2.4 Priors and Distributions

The prior distributions are selected as follows.

$$P(\phi) = \frac{1}{2\pi}, -\pi < \phi \leq \pi \tag{20}$$

$$P(r_i) = \frac{1}{r_{max} - r_{min}}, r_{min} \leq r_i \leq r_{max} \tag{21}$$

$$P(a_i) = \frac{1}{a_{max} - a_{min}}, a_{min} \leq a_i \leq a_{max} \tag{22}$$

$$P(O_i) = (1 - p_{oc}, p_{oc})^T \tag{23}$$

p_{oc} is the prior probability that a a contour is an occluding contour.

Especially the prior for r_i plays an important role, since it will help to sift out a certain class of contours that fit the shading model well but are not occluding contours. This class are contours which have a flat intensity curve. They come typically from shadows or the outsides of object boundaries on planar surfaces. Because of the flat intensity curve their estimated r_i will be close to zero and can then be easily sifted out by setting r_{min} to a value over zero.

The estimates, as derived in Section 2.2, are normally distributed around the true values, provided that the contours are occluding contours. The estimates from other contours are also normally distributed, for the same reason as the occluding contours, but around which mean $\vec{m}$ is unknown. The estimates certainly doesn't tell us anything about the variables we are estimating. From (4) it can be shown that the mean is

$$\vec{m} = E\left[(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{I}\right] = \mu_I \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad (24)$$

where μ_I is the expected value of the image intensities. By calculating a mean over a number of images μ_I was estimated to 0.35. A suitable covariance matrix was roughly estimated to $S = ((0.5, 0, 0)^T(0, 0.5, 0)^T(0, 0, 0.5)^T)$

Hence, the distribution of the estimates is modeled by

$$P(\hat{L}|r_i, \phi, a_i, O_i) = \\ = \begin{cases} g(\hat{L}, (r_i\cos\phi, r_i\sin\phi, a_i)^T, \hat{\Sigma}_i) & O_i = oc \\ g(\hat{L}, \vec{m}, S) & O_i = \bar{oc} \end{cases} \quad (25)$$

The function $g(\vec{x}, \vec{\mu}, \Sigma)$ is the three-dimensional gaussian p.d.f. with mean $\vec{\mu}$ and covariance Σ.

3.2.5 Implementation Issues

The Bayesian network in Figure 2 is a hybrid network, meaning it has both continuous and discrete variables. Although we have normal distributions we can not use the developed techniques in e.g. [2]. This is because the argument ϕ is a common variable in the cliques which messes up the marginalizations. Fortunately many of the marginalizations can be solved analytically. Maximizing over r_i and a_i is done by minimizing the quadratic expression in the exponential of the gaussian. With limits on r_i and a_i we need to check the boundaries of the limit region as well.

When maximizing over O_i we need to calculated the probability distributions numerically and therefore it is necessary to discretize ϕ.

4 Experiments

All the experiments are done on images captured on an Olympus 3030-Z digital camera. The correct light source direction was measured using the shadow of a small sphere on a piece of wire placed in the scene.

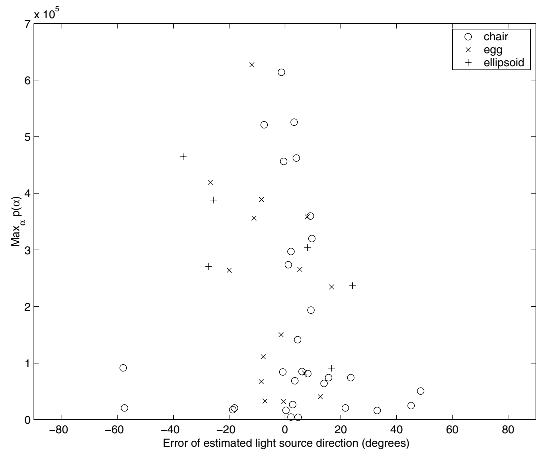

Figure 4. Error of light source direction estimated from occluding contours. plotted versus the maximum of the probability density.

4.1 Light Source from a single Occluding Contour

To test the estimation of light source direction from a single occluding contour we used in total 41 contours from three different objects illuminated from six different directions. The contours where extracted by manually selecting parts of contours extracted by the Canny edge detector followed by edge linking.

In Figure 4 the errors of the estimated light source direction is plotted versus the maximum probability density of the estimated vector. The maximum probability is relevant because it plays an important role in the message propagation in the Bayesian network. Some of the estimates contain non-negligible errors, but it can be seen that the computed variance reflect the errors, which is of great importance when fusing them.

4.1.1 The Benefits of Extrapolation

What are the benefits of using extrapolation? The alternative is to use the intensity value closest to the contour. For many objects this would be fine. It can be seen in the shading model (1) that we can include contour points having a normal with non-zero z-component, as long as the z-component is constant along the contour for which the points are measured. Since the term kn_zl_z is constant it can be fit into the simpler model (2) where the term a actually is $a + kn_zl_z$.

Contours on objects which have more or less a constant cross-section radius will therefore not benefit from the extrapolation. Instead the extrapolation will just amplify the noise. The more complex extrapolation model that is used

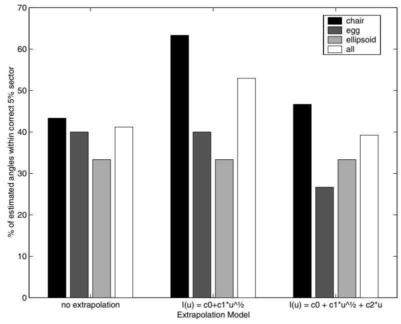

Figure 5. The number of estimated angles within the 5% circle sector around the correct angle for different number of terms in the extrapolation model.

the more the noise will be amplified. In our experiments using an object shaped as an egg or an ellipsoid we will typically not benefit from the extrapolation. The cross-section will be more or less constant along the occluding contours of these object. A chair however have a more varied contour and should therefore benefit from the extrapolation.

This can be seen in Figure 5, which is a graph of the percentage of the estimated angles that are within 5% sector around the correct angle, for different extrapolation models. When using $I(u) = c_o + c_1 u^{1/2}$ as opposed to doing no extrapolation[2] we see no increase in the performance of the egg and the ellipsoid. The chair on the other hand has an increase from 43% to about 64%. As more terms are added to the extrapolation model the amplification of the noise affects the results and reduces the performance, see Figure 6.

4.2 Automatic Light Source Estimation

To test[3] the automatic algorithm two sets of images were used. The first set of images was designed to be easier by having a number of objects with occluding contours in the image. Also, the objects where placed apart so that the grouping algorithm would have less problems finding candidate contours. The set contains 14 images of two scenes illuminated from different directions. For each image the light source direction was estimated using the automatic algorithm. Also a probability measure telling how much the estimate can be trusted was calculated. This was done by summing up the probabilities for all ϕ within $\pm 5°$ of the estimated angle $\hat{\phi}$, thereby making it invariant to the number of discretization levels of ϕ. Note that the message propagation produces the whole distribution for ϕ. Figure 7 shows

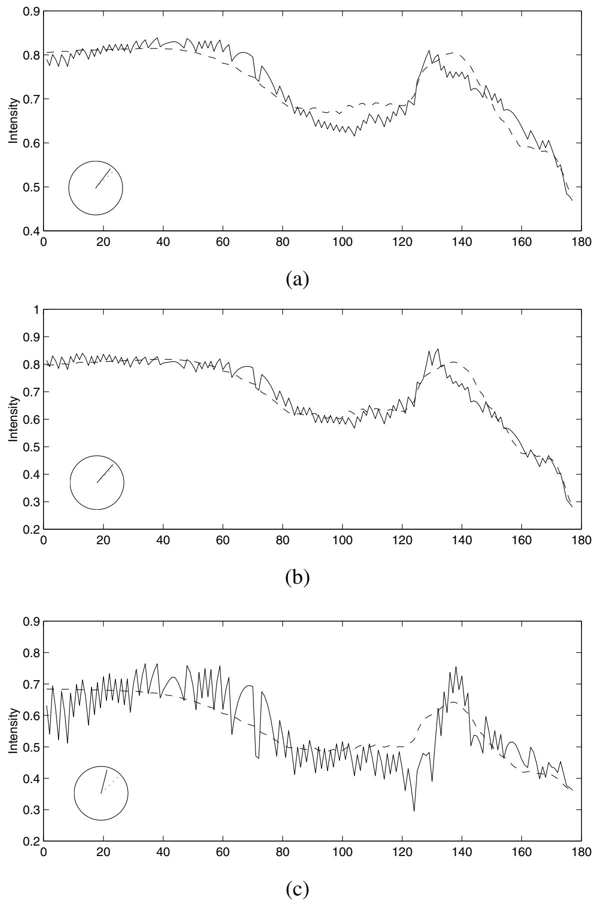

Figure 6. The image intensities along one of the occluding contours in the dataset. The solid lines show the extrapolated intensities and the dashed lines the reconstructed intensities from the estimation, when using extrapolation models a) no extrapolation, b) $I(u) = c_0 + c_1 u^{1/2}$ and c) $I(u) = c_0 + c_1 u^{1/2} + c_2 u$. The inlined circles show the estimated and correct (dotted line) light source directions.

the error of the estimated angles plotted versus the probability measure. As many as half of the estimated angles have less than 5° error. There are also outliers, such as one with 33° error and a very high probability. In the second set of images the objects sometime occlude each other. The set contains 17 images. Figure 8 shows the results. The algorithm is able to estimate light source direction well, with some outliers.

The outliers occur for different reasons. Sometimes a non-occluding contour fits the shading model well. In other cases there are many non-occluding contours accidentally giving similar estimates and thereby reinforcing each other. This mainly happens in the absence of a correct occluding contour giving a good estimate. When the grouping algorithm successfully picks out a good occluding contour,

[2]When no extrapolation has been used, the image intensity is just measured two pixels from the contour.

[3]The parameters in the priors were in all experiments set to: $a_{min} = 0$, $a_{max} = 0.6$, $r_{min} = 0.1$, $r_{max} = 1$ and $p_{oc} = 0.1$.

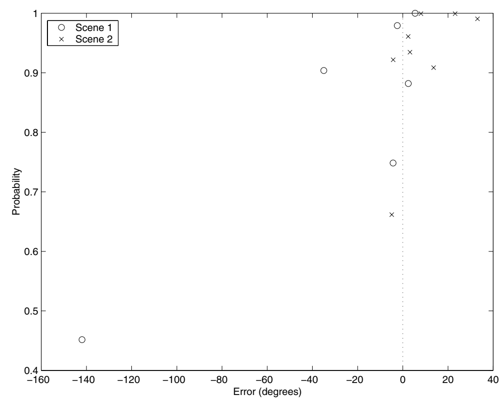

Figure 7. Error of estimated light source direction plotted versus probability for first image set.

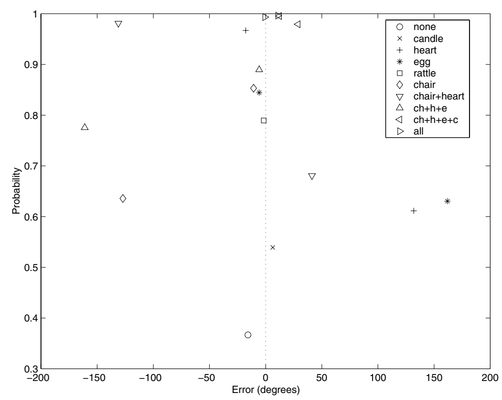

Figure 8. Error of estimated light source direction versus probability for second image set.

this contour will usually give a better estimate than non-occluding contours.

Note that the probabilities in this case should be compared relatively and not be considered as good estimates of the true probabilities. With better modeling, using e.g. learning, the algorithm should in the future be able to produce probabilities that better reflect the true values.

Figure 10 shows two examples of the algorithm estimating the light source direction and classifying the contours. In the first example the direction is correctly estimated, but some of the contours are incorrectly classified as occluding. This can happen when a contour accidentally gives the same estimate as the "winning" light source direction. In the second example the algorithm fails to estimate the correct light source direction and thereby fails to classify the contours.

5 Conclusion

We have presented a way to estimate the projected light source direction. The algorithm exploits the occluding boundary. Combined with a heuristic grouping algorithm and Bayesian network inference this is done in a fully automatic way. Contours picked out by the grouping algorithm are also classified as occluding or not.

Our approach to estimate the light source direction from a single occluding contour has been tested on real images with good results. The experiments show that the calculated variance reflects the errors of the estimates, which is crucial when fusing the estimates.

The automatic algorithm shows promising results. A first set of images test the inference stage on scenes suitable for the grouping algorithm. A second image set of more challenging scenes causes the grouping algorithm to fail more often. The algorithm as a whole still works with reasonable robustness.

Future improvements can be achieved in two main respects. The grouping algorithm should incorporate some sort of good continuation to better be able to cope with edges in the background. Moreover, one could model the distribution of estimates from non-occluding contours better, e.g. by learning the distributions.

References

[1] P. Belhumeur, D. Kriegman, and A. Yuille. The bas-relief ambiguity. In *Proc. CVPR*, pp 1060–1066, 1997.

[2] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[3] B. Horn. *Robot Vision*. McGraw-Hill, 1986.

[4] F. Jensen. *An Introduction to Bayesian Networks*. UCL Limited, 1996.

[5] G. Klinker, S. Shafer, and T. Kanade. A physical approach to color image understanding. *IJCV*, 4(1):7–38, January 1990.

[6] M. Oren and S. Nayar. Generalization of the lambertian model and implications for machine vision. *IJCV*, 14(3):227–251, April 1995.

[7] A. Pentland. Finding the illuminant direction. *JOSA-A*, 72:448–455, 1982.

[8] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proc. ICCV*, pp 875–82, 1999.

[9] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from shadows. In *Proc. CVPR*, pp 306–12, 1999.

[10] E. V. Vega and Y.-H. Yang. Default shape theory: With the application to the computation of the direction of the light source. *JOSA-A*, 60:285–299, 1994.

[11] L. Wolff. Diffuse-reflectance model for smooth dielectric surfaces. *JOSA-A*, 11(11):2956–2968, November 1994.

[12] Y. Yang and A. Yuille. Sources from shading. In *Proc. CVPR*, pp 534–539, 1991.

[13] Y. Zhang and Y.-H. Yang. Illumination direction determination for multiple light sources. In *Proc. CVPR*, pp 269–76, 2000.

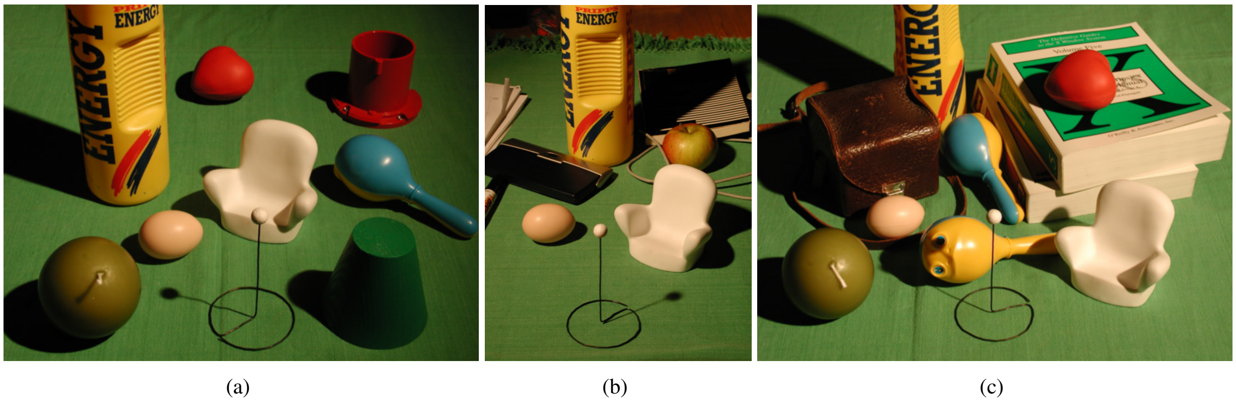

Figure 9. Some of the images used in the experiments.

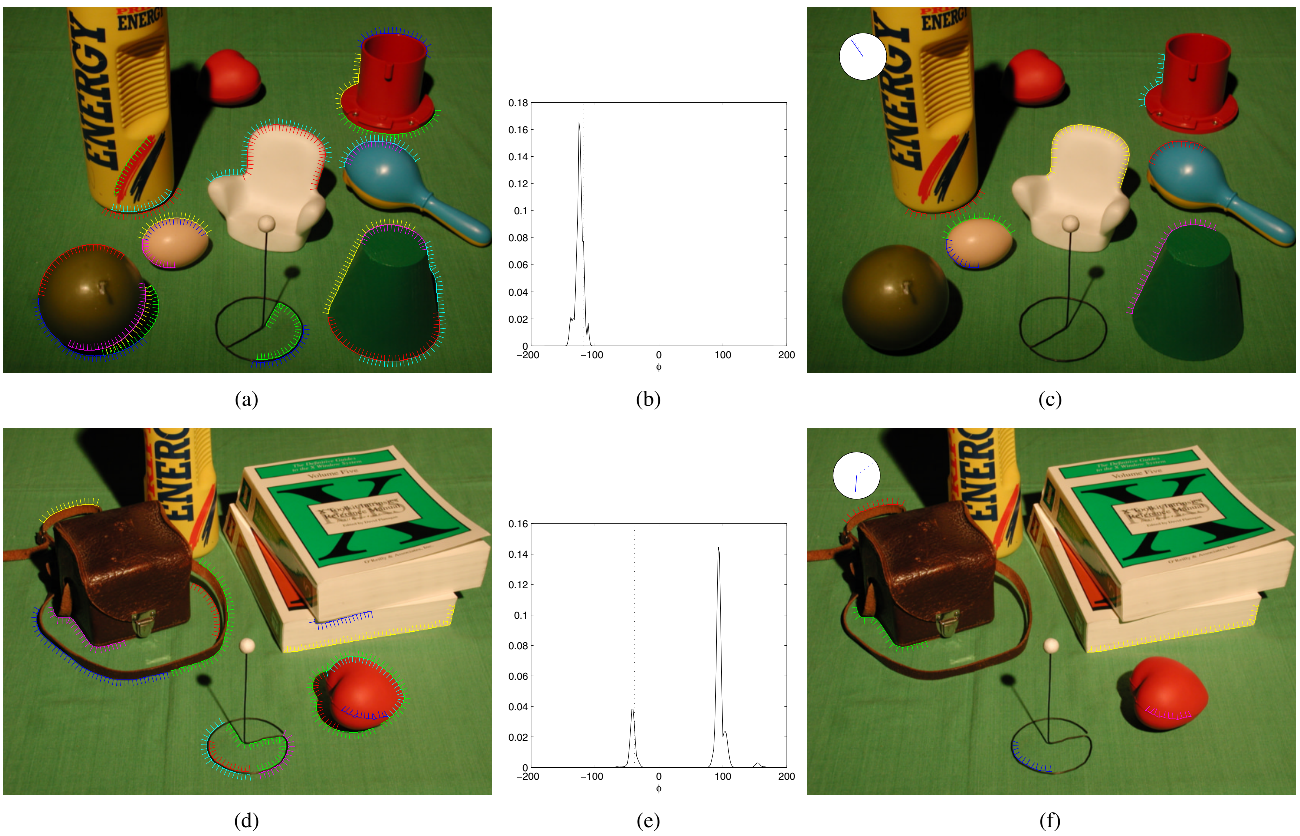

Figure 10. Two examples: a, d) Candidate contours from the grouping algorithm. b, c) probability distribution of estimated angle after message propagation. c, f) Contours classified as occluding. The inlined circle shows the estimated angle. Dotted lines show the correct angles.