Finding Glass

Kenton McHenry, Jean Ponce
Beckman Institute
University of Illinois
Urbana, IL 61801, USA

David Forsyth
Department of Computer Science
University of Illinois
Urbana, IL 61801, USA

Abstract. *This paper addresses the problem of finding glass objects in images. Visual cues obtained by combining the systematic distortions in background texture occurring at the boundaries of transparent objects with the strong highlights typical of glass surfaces are used to train a hierarchy of classifiers, identify glass edges, and find consistent support regions for these edges. Qualitative and quantitative experiments involving a number of different classifiers and real images are presented.*

1 Introduction

This paper address the problem of finding glass objects in images, focusing on the identification of their internal and external image contours in the output of an edge detector. Typical glass objects such as bottles, glasses, plates, and vases have rather smooth surfaces, display strong highlights but weak diffuse reflectance, and—as pointed out by Murase [16]—reveal a distorted version of the texture of surfaces lying behind them (Figure 1). These properties are used in this paper to identify a number of visual cues associated with the image regions surrounding glass edges. Given a set of training images with hand-labeled glass edges, we then use these cues to learn a hierarchy of classifiers and identify fragments of glass boundaries in new pictures (see [12] for related work in the image segmentation domain). A global integration step merges these fragments and uses snakes to identify their support regions as potential glass objects.

1.1 Background

The reflective and refractive properties of transparent objects can be used to recover their shape from images [2, 3, 8, 15, 16], or identify known three-dimensional shapes in photographs [18]. They can also be used to identify reflections in transparent surfaces and separate them from the background: Given the motion of the two layers, [22] uses a linear algorithm to recover the two parts. Levin et al. [10] makes the observation that this can be done by separating an image into two images such that the number of edges and corners is minimized. Adelson and Anandan [1] introduce a linear model for the intensity of a transparent surface, namely

$$I = \alpha I_B + e \tag{1}$$

where I_B is the intensity of the background, α is a a blending factor, and e is the emission of the (semi) transparent surface itself. They use this model to derive a set of constraints on the brightness patterns of X-junctions at the boundary of transparent objects. From these a junction can be classified as one of three types: non-reversing if the intensity shift is in the same direction along both edges of the X, single reversing if it switches along one edge, and double reversing if it switches along both edges. Both non-reversing and single-reversing junctions allow for the possibility of a transparent edge being present, but double-reversing ones do not. Singh [20] uses these constraints to separate transparent overlays from their background: Regions with similar color and texture are clustered together, and their edges are found. Each of the edge X-junctions are then labeled as either non-reversing, single-reversing, or double-reversing. The connected edges are then followed until returning to the junction, in effect outlining a region covered by a transparent overlay, or running into a double-reversing junction. If a double-reversing junction, is reached the algorithm backtracks to any single-reversing junctions that were seen along the way to try a different route. The reason for the clustering is to reduce the amount of data and increase the opportunity of successfully separating transparent regions. Notice that the constraints do not guarantee a transparent edge is present, but only the possibility. In fact it is very likely that an albedo pattern may satisfy these constraints. Thus taken locally the constraints only offer an indication of a possible transparent edge and must be then looked at globally to see if a region can be found that agrees with all the responses.

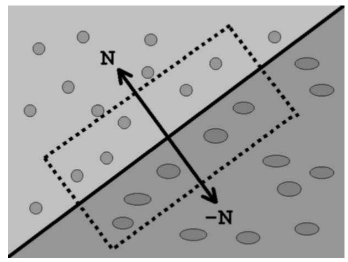

Figure 1: *An edge separating two regions. To determine if the edge is caused by a glass surface we examine a small part of the edge and compare the patches on both sides. If the edge is from a glass object then both sides should be similar and distorted. The distortion is required in order to distinguish the edge from other edges within textures.*

The method proposed in this paper does not rely on the detection of X-junctions, which is difficult in realistic imagery, but it does exploit the linear model of Eq. (1).

2 Edge Cues

Given contour fragments from an edge detector we would like to identify those that are associated with glass. Kaufhold and Hoogs [12] use various cues to remove edges that a person would not consider to be true object boundaries. Basically, the goal is to keep edges that are between two non-similar regions. Here we are not concerned with true object boundaries but instead with internal and external edges of glass regions. Since glass is usually clear, we focus on its refractive/reflective properties. Assuming that a sample of the background is visible and undistorted on one side of a glass edge, we identify five cues to the presence of such an edge.

- **Color Similarity:** the color tends to be similar on both sides (Section 2.1).
- **Blurring:** the texture on the glass side is blurrier (Section 2.2).
- **Overlay Consistency:** the intensity distribution on the glass side is constrained by intensity distribution on the non-glass side (Section 2.3).
- **Texture Distortion:** the texture on the glass side is slightly different (Section 2.4).
- **Highlights:** the glass side has a specularity (Section 2.5).

Note that internal edges in glass regions often correspond to sudden changes in object thickness or shape, and one can still consider that the texture on one side of the edge is a distorted version of the texture on the other side.

Each of the fives cues used in this paper is characterized by one or two scalar values that provide information relevant to glass properties. Similar color distributions and high alpha values indicate possible transparency. Blurring and distortion deal with refractive properties, while emission and highlights deal with reflective properties. Below we describe how the values for each cue are obtained.

2.1 Color Similarity

The value returned provides an indication of how similar the colors are across the edge. To measure this a histogram is constructed for each side containing twenty bins for range of hue values and twenty bins for the saturation values. The intensity component is ignored in order to make the measure somewhat invariant intensity differences. The two histograms are normalized to have a sum of one and the euclidean distance between them calculated. This distance is the value returned.

2.2 Blurring

In our experiments it has been observed that the side covered by glass is often a smoother version of the other side. We explain this as the result of impurities in the glass causing imperfect refraction as well as minor surface variations which create different optical properties along the surface. Thus the value returned by this cue should indicate how much smoother one side is compared to the other.

The method for measuring texture smoothness proposed by Forsyth and Fleck [7] involves subtracting a smoothed version of image from the original. Already smooth areas will have small differences where as highly textured areas will have large differences. To measure the smoothness of a 2D sample we turned to the discrete cosine transform. Once the two sides are transformed we compute the mean of the frequency coefficients on each side. The difference among the two means can then be used to indicate relative smoothness (as being a shift into the lower frequency range of the spectrum). For this measure to make any sense we must consider how textured the background was to begin with. A small shift on a highly textured background could just be noise while on a smooth background is much more significant. We thus normalize the measured shifts by a measure of the texture entropy which we take to be the standard deviation of the intensities on that side of the edge.

2.3 Overlay Consistency

Rather than using the constraints of [1] we directly use the model given in Eq. (1). The two parameters α and e can be used to identify edges exhibiting transparent overlay effects. An α value near 1 and a low value of e indicates that the edge is likely an intensity change in a textured region. An α less than 1 with and small e may indicate that the edge belongs to a transparent object. A low α and/or high e indicates a strong intensity change indicating an edge between two very different regions. Thus these two parameters will make up the values of this cue.

Equation (1) is a linear in two unknowns, and can thus be solved given at least two distinct intensities on each side of the edge. This is interesting and makes sense since given a homogeneous background there is no reason to assume the presence of a transparent overlay rather than just a change in intensity. Since Eq. (1) is a 1D affine transformation resulting in only a scaling and translation (and assuming the texture is consistent on both sides of the glass edge and that our sample is large enough) we can assign I and I_B by clustering the intensities on the two sides of the edge by percentages. In other words we can say that the top 10% brightest on one side is equal to the top 10% on the other side and so on. The mean intensity of the clusters of one side becomes the values of I_B in Eq. (1) and the means of the clusters on other side the values of I. The value of α and e can now be solved as a linear least squares problem. The value of α should be between 0 and 1, so if it is negative we simply switch the values of I_B and I.

One may ask that since glass is clear shouldn't α always be 1 and e always be 0 for glass edges. In practice the α value actually tends to be a bit less than 1 on glass edges, again likely due to the smoothness phenomena. Also, the emission value appears to correspond with weak reflections in the glass.

2.4 Texture Distortion

Because of refraction we can expect that a textured background will be magnified and/or skewed when seen through glass. The value returned from this cue should provide information as too how similar the texture is on both sides of the edge.

To measure the difference in texture we filter both sides of the edge with a bank of gaussian filters consisting of six orientations and two scales [14]. Distributions of the filter outputs are created for both sides of the edge. The similarity of the texture on the two sides is then measured as the euclidean distance between the two distributions.

2.5 Highlights

Glass is known to be highly specular, making highlights a valuable cue. The value returned from this cues is a binary value indicating if a highlight was detected on one side of the edge.

As discussed by Klinker et al. [13] highlights can be found in color images by assuming a dichromatic color model and looking for dog leg like structures in a plot of the colors observed in

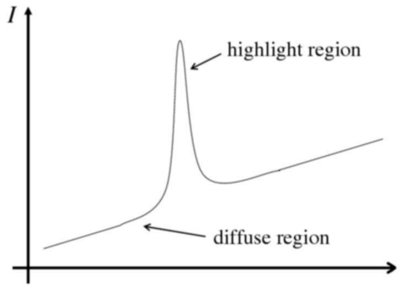

Figure 2: *Typical profile of a highlight on a smooth specular surface.*

small windows. While an effective technique in areas containing monochrome surfaces, searching for the dog leg structures in textured areas can be a nuisance. Since highlights are usually the brightest part of the image (being reflections of the light source) a simple technique is to threshold based on intensity (e.g. keeping pixels above 90% of the brightest value). However we can do better. Clearly it is desirable to set the threshold as low as possible in order to get as much of the highlight as possible. At the same time we do not want to introduce bright patterns resulting from non-specular reflections.

We use the following heuristic to detect highlights in a reliable fashion. First note that highlights on smooth shiny surfaces such as those of most glass objects tend to have a profile such as the one shown in Figure 2, where a sharp spike is overlaid on a (locally) smooth intensity profile due to diffuse reflection and/or, in the case of transparent objects, background texture. Various analytical models for the shape of this spike exist, from the non-physical Phong model commonly used in computer graphics, to physical models related to the microscopic roughness of the surface, such as those proposed by Healey and Binford [9] and Nayar et al. [17]. For our purposes, it is sufficient to note that, on each side of the spike and over most of its (small) extent, the intensity profile is close to a straight line. In turn, this means that the perimeter P of a two-dimensional highlight region found by thresholding should also be a roughly affine function $P = aT + b$ of the threshold T. The perimeter of a bright diffuse region, on the other hand, tends to be (roughly) piecewise constant.

Thus to find the proper highlight threshold we create a plot of highlight perimeter as a function of intensity threshold. Once an image has been thresholded, such that pixels below the threshold are black and pixels above are white, we estimate the perimeter simply by counting the number of sides on each white pixel that face black pixels. As can be seen in the middle plot of Figure 3 the tip of a spike can be seen lying on its side near the end of the plot. As argued above, we can approximate the edge of the spike by a straight line. Thus to find the threshold we iteratively fit a line to the perimeter values , starting from a threshold of 1.0 and plot the fit error (see Figure 3). As can be seen the error is nearly 0 until the bottom of the spike region is hit (indicated by an arrow in the figure). An example is shown in Figure 4 where the threshold is set to 0.71 for a glass plate. Had the threshold been set to a fixed value of 90% of the brightest intensity, a value of 0.87 for this image, much of the highlights would have been lost.

By setting the threshold in this way we can set it as low as possible to get as much of the highlight without worry of including

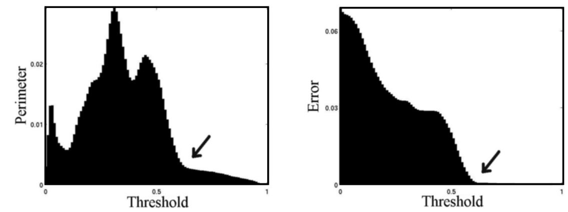

Figure 3: *The relationship between intensity threshold and perimeter of iso-intensity rings within highlights.* **Left:** *Plot of perimeter as a function of the intensity threshold.* **Right:** *Plot of line fit error to perimeter in the previous plot as a function of the intensity threshold. The intensity at which the error is no longer small is chosen as the threshold to determine highlights. The chosen threshold of 0.71 is indicated by an arrow.*

mostly bright Lambertian regions. For example if a white sheet of paper is placed in the image at an appropriate angle as to be fairly bright, the threshold would be set just above its maximum brightness. While this is what we want it is undesirable for this to suppress highlights through out the image. We overcome this by examining sub-windows instead of the whole image at once.

3 A Local Classifier

Each of the parameters, except for highlights which are either 0 or 1, should occupy a small range of values at a glass edge: color differences should be small, smoothness shifts small but non-zero, alphas near one, emissions low, and distortion somewhat high. To learn the space of parameters occupied by glass we use a support vector machine with a gaussian kernel. The result is a hyperplane in the space described by the kernel function that maximizes the margin between the positive and negative data.

We must be very conservative when labeling an edge as glass. There are many situations in which non-glass edges are similar to glass edges as defined by the cues. One example is a lined surface such as the mat shown in Figure 7. Across the lines there is a similar color distribution, the possibility of a low alpha value and non-zero emission value (due to shading). Something similar could be said for shadows. However, we have another cue in that we know the glass in the image encloses a region. Thus if the precision of the classifier is somewhat high we can use a global integration step to link sparsely found glass edges together (Section 5). Thus our goal when training the classifier should be to minimize the number of false positives.

A method of minimizing false positives is to have a large number of negative examples and weighting the negative data in the training set more than the positive data. Thus to discourage calling such edges glass we can collect a large collection of nega-

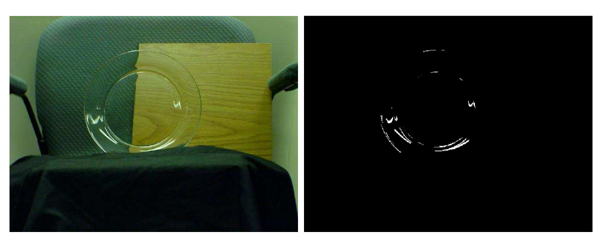

Figure 4: *A glass plate and its detected highlights at a threshold of 0.71.*

tive samples embodying as many cases of such undesirable edges as possible and then give them a high weight. Even with a large weight forcing no false positives on the training data the learned classifier tends to have too many false positives on test data. We encounter this problem due to our quantized view of the space. We would like the number of false positives to be extremely low, perhaps one in one million edge samples. In order to reach such a rate we would have to acquire an amount of negative samples that would be impractical to train a classifier with.

A more practical option to limiting false positives on test images is to set the recall rate low enough such that only the strongest true glass edges are labeled as glass and the weaker glass edges along with these false glass edges are labeled as non-glass. To do this we turn to another technique to adjust the recall of an SVM which is to adjust the position of the hyperplane away from the negative data. In other words we shift the constant parameter so that the decision boundary passes into the positively labeled data, increasing the number of negatively labeled data. We do this until the number of true positives drops below 30% on the training data.

4 Multiple Classifiers

Let us consider the range of values for each of the cues parameters in glass as logical propositions. From this we make the observation that for an edge to be glass only two of these propositions must always be true given that one of the other four is true as well:

$$\begin{aligned} glass &\Leftarrow similar\_color \wedge high\_alpha \wedge \\ &\quad (low\_emission \vee highlight \vee \\ &\quad smoother \vee distortion) \end{aligned}$$

The above statement can be re-written as four different statements of three propositions:

$$\begin{aligned} glass &\Leftarrow similar\_color \wedge high\_alpha \wedge low\_emission \\ glass &\Leftarrow similar\_color \wedge high\_alpha \wedge highlight \\ glass &\Leftarrow similar\_color \wedge high\_alpha \wedge smoother \\ glass &\Leftarrow similar\_color \wedge high\_alpha \wedge distortion \end{aligned}$$

The underlying structure comes from the fact that the color and alpha parameters are sufficient in eliminating most non-glass edges. What is left is to distinguish among the two types of edges that satisfy the proposition associated with these parameters, internal texture edges and glass edges. Thus we need information from at least one, not all, of the other cue parameters.

Since the problem has this structure it may prove beneficial to use it and learn four classifiers in the lower three dimensional spaces of the four statements above, rather than a single classifier using all six cue parameters. To combine the outputs of these classifiers we consider the three methods described below.

4.1 Logical OR

The most obvious means of combining the classifiers is to OR their outputs. That is if one classifier labels an edge as glass then the edge will be labeled glass.

4.2 Weighted Sum

An alternative to OR'ing the classifier outputs is to weight them and see if their sum is above some threshold. By combining the cues in this way we are able to give less weight to cues that do not perform as well as the rest. It may also be the case that sets of cues labeling an edge as glass is better than considering each one separately. If this is the case then no one weight will be above the threshold.

One method of finding the weights and threshold is to use gradient ascent to maximize the classifier's accuracy on the training set. However, one should notice that the parameters we are looking for define a support vector classifier with linear boundaries.

4.3 Exponential Model

One can combine boolean classifiers so as to return a probability of a given label given their outputs. Maximum likelihood is used to learn the parameters of an exponential model, whose form is given by entropy considerations as:

$$p(y|x) = Z(x)e^{\sum_i \lambda_i f_i(x,y)}$$

where $p(y|x)$ is the probability of a label y being assigned to the SVM classifier outputs x [4]. λ_i is a weight for a particular feature defined by the function $f_i(x, y)$. These feature functions return a value of 1 if x takes a particular form given y and 0 otherwise. $Z(x)$ is a normalizing factor. The parameters are found with generalized iterative scaling [6]. Once the parameters are found we can classify a particular set of sub-classifier outputs by thresholding on a probability of 0.5.

5 Global Integration

Once a part of an edge has been classified as glass we should more easily believe that the rest of the edge is glass as well. We can increase the number of positive results by connecting two edge samples labeled as glass if they have a path connecting them. As will be described in the following section, edges from the edge detector are broken into smaller edge samples for classification. Some of these may be labeled as glass while some may not, even if the whole edge is glass. These false negatives can occur if the assumption is locally violated (i.e. the background is not the same on both sides) or because of the conservative training of the classifier. This hysteresis style of approach provides us with a means of recovering these falsely labeled glass edge samples.

We now have a number of long edges that are labeled as glass. To find the glass region that these edges enclose we can use an active contour approach such as the one described by Kass et al. [11]. Initialized at the boundary of the image a polygon is iteratively updated so as to minimize an internal energy function enforcing smoothness and an external energy function preferring to be near edges. Once the contour has converged all pixels within its boundary are labeled as glass.

6 Experiments and Results

Given an image we find edges with the Canny edge detector. From an edge point we then walk 25 pixels in each direction, and using the center pixels normal we take a sample 25 pixels outward in the positive and negative direction. The edge samples are then transformed into a 6-vector using the described cues. The results from the various cues can be seen in Figure 6.

To train the classifier 15 images were used. Six of these images contained glass objects in front of various backgrounds. Edges

Classifier	True Positive Rate (Recall)	False Positive Rate	Precision
Single SVM	47.01%	3.09%	68.76%
Multiple SVM's + OR	88.30%	10.04%	56.04%
Multiple SVM's + Weighted sum	83.94%	8.53%	58.78%
Multiple SVM's + Exponential model	88.30%	10.04%	56.04%
Multiple SVM's + Weighted sum (sampled)	79.72%	4.12%	73.7%

Table 1: *Results from the described classifiers, all tested on a test set of 50 images where glass pixels were marked by hand. The true positive rate, or recall, indicates the percentage of glass pixels that were correctly identified. The false positive rate indicates the percentage of non-glass pixels that were identified as glass. Precision is the percentage of identified glass pixels that are actually glass.* **From top to bottom***: an SVM trained on all six values of the cues, a classifier consisting of the OR'ed output of the four sub-classifiers as described in section 4.1, a classifier consisting of a weighted sum of four sub-classifiers as described in section 4.2, and a classifier consisting of the outputs of four classifiers combined through an exponential model as described in section 4.3. The last row contains results from a classifier consisting of the weighted sum of four sub-classifiers trained on a subset of the training data as described in section 6.*

were manually labeled and broken into 333 positive training samples. The other nine images contained no glass. An edge detector provided the edges which were then broken into 4581 negative samples. To test the various classifiers a set of 50 images was used. Thirty five of these images contained glass objects in front of various backgrounds. Glass regions were manually labeled and stored in a separate image mask. The remaining fifteen images contained non-glass objects.

To measure how well the classifiers found glass each one was run on the set of 50 hand labeled test images. The regions identified as glass by the classifier are then compared with the masks associated with each image. True positives are measured as the number of pixels within the intersection of the two regions while false positives were measured as the number of pixels within the snake but outside the mask. The results can be seen in Table 1. The classifier trained using the six cues has a recall of 47% and a precision of 68%. The classifiers made of sub-classifiers have higher recall rates, all around 80%, but at a somewhat lower precision.

In addition to the four classifiers discussed before we experimented with the idea of training a classifier on subsets of the training data rather than the whole set. The subsets were chosen randomly so that we had 50 positive samples and 100 negative samples. A classifier trained on such a subset was considered good if it could correctly classify the edges on the difficult image discussed earlier of a glass plate on a lined mat. The best of these classifiers is shown in the last row of Table 1. This classifier out performed the others with a high recall rate as well as a high precision.

Six examples where the glass was successfully identified are shown in Figure 7. Figure 5 shows an example of where the classifier falsely labels edges as glass. The falsely labeled edges are the result of the large texture pattern in the image. Since we are looking at each side of the edge through a window of finite size we fail to get accurate statistics from larger texture patterns. In this example the color is similar on both sides of the edges along the internal of the brick pattern. The alpha value is likely high as well. Due to our limited view, the texture appears to be different on each side of the edge, thus these non-glass edges are labeled as glass. It should be pointed out that in this image there are a significant amount of edges within the brick that were correctly classified as non-glass.

Figure 5: *Test image showing edge samples that are falsely labeled by the local classifier (shown in white).*

7 Conclusion

We have presented a method for identifying glass edges and regions in an image. It is not limited to glass since most of the cues we have used are valid for other smooth transparent media. It does, however, assume that the background is similar on both sides of all glass edges. This is of course not true when refraction is too strong, or when transparency is overwhelmed by very bright highlights.

Acknowledgements. This work is partially supported by the National Science Foundation under grant IIS-0308087, Toyota Motor Corporation, and the Beckman Institute.

References

[1] E. Adelson, P. Anandan. "Ordinal Characteristics of Transparency," *AAAI*, pp. 77-81, 1990.

[2] S. Agarwal, S. Mallick, D. Kriegman, S. Belongie. "On Refractive Optical Flow," *ECCV*, pp. 483-494, 2004.

[3] M. Ben-Ezra, S. Nayar. "What Does Motion Reveal About Transparency?," *ICCV*, pp. 1025-1032, 2003.

[4] A. Berger, S. Pietra, V. Pietra. "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, pp. 39-71, 1996.

[5] G. Brelstaff, A. Blake. "Detecting Specular Reflections Using Lambertian Constraints," *ICCV*, pp. 297-302, 1988.

[6] J. Darroch, D. Ratcliff, "Generalized Iterative Scaling for Log-linear Models," *The Annals of Mathematical Statistics*, pp. 1470-1480, 1972.

[7] D. Forsyth, M. Fleck. "Finding Naked People," *ECCV*, pp. 593-602, 1996.

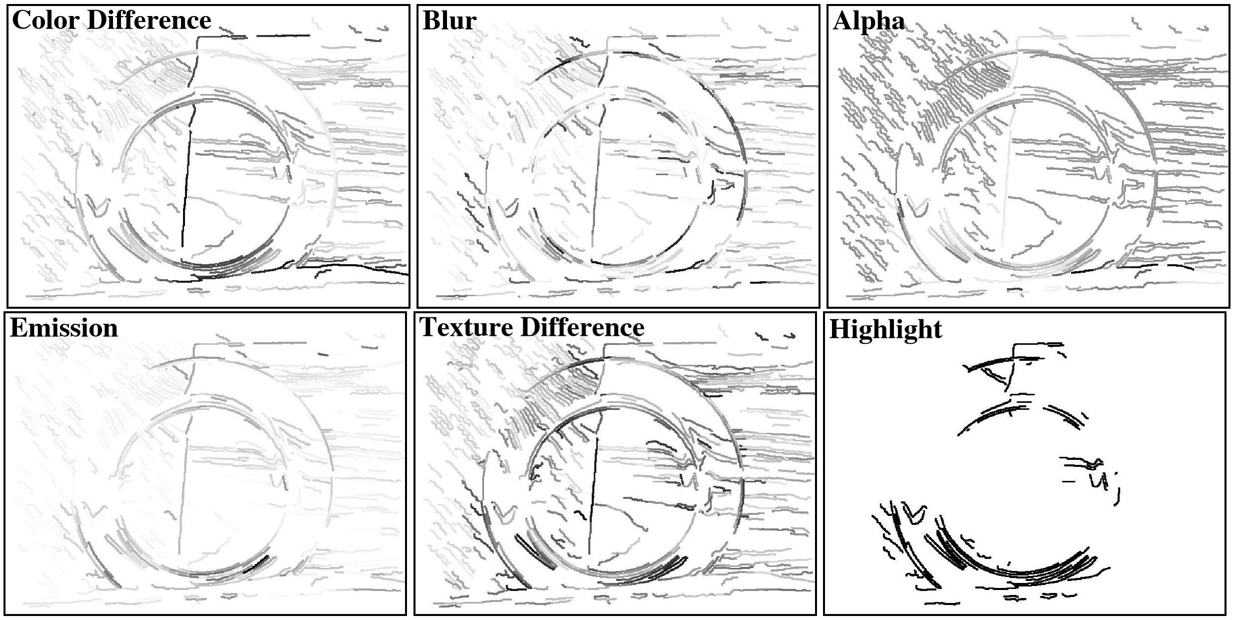

Figure 6: *The values provided by the six cues on the image of a glass plate in Figure 4. In each image darker edges indicate higher values.* **Top left:** *The difference in the color distributions across the edge as measured by the method in section 3.1. Notice the highest value along the edge of the wooden board and fabric behind it which has a very different color distribution.* **Top middle:** *The amount of blurring across the edge as measured by the method in section 3.2. There are few edges with large blurring values within consistent texture regions.* **Top right:** *Alpha values across edges as measured by the method of section 3.3. The edge of the board and fabric is not visible since being very different in intensity distribution the alpha value here is very low.* **Bottom left:** *Emission values across edges as measured by the method in section 3.3. Again the edge of the wooden board is visible as well as much of the glass plate.* **Bottom middle:** *Measure of difference in texton distributions as measured by the method of section 3.4. Notice here that the lowest values exist within a consistent texture.* **Bottom right:** *Highlights, a binary value determined by the method of section 3.5. A value of one exists along edges that have a detected highlight to one side.*

[8] S. Hata, Y. Saitoh, S. Kumamura, K. Kaida. "Shape Extraction of Transparent Object Using Genetic Algorithm," *ICPR*, pp. 684-688, 1996.

[9] G. Healey, T. Binford. "Local Shape From Specularity," *Proc. Image Understanding Workshop*, volume 87, pp. 874-887, 1987.

[10] A. Levin, A. Zomet, Y. Weiss. "Separating Reflections from a Single Images Using Local Features," *CVPR*, 2004.

[11] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models." *IJCV*, volume 1, pp. 321-331, 1987.

[12] J. Kaufhold, A. Hoogs. "Learning to Segment Images Using Region-Based Perceptual Features," *CVPR*, volume 2, pp. 954-961 2004.

[13] G. Klinkir, S. Shafer, T. Kanade. "A Physical Approach to Color Image Understanding," *IJCV*, volume 4, number 1, pp. 7-38, 1990.

[14] J. Malik, S. Belongie, T. Leung, J. Shi. "Contour and Texture Analysis for Image Segmentation," *ICCV*, pp. 7-27, 2001.

[15] D. Miyazaki, M. Kagesawa, K. Ikeuchi. "Polarization-based Transparent Surface Modeling from Two Views," *ICCV*, pp. 1381-1386, 2003.

[16] H. Murase. "Surface Shape Reconstruction of an Undulating Transparent Object," *ICCV*, pp. 313-317, 1990.

[17] S. Nayar, K. Ikeuchi, T. Kanade. "Surface Reflection: Physical and Geometrical Perspectives," *PAMI*, volume 17, number 7, 1991.

[18] M. Osadchy, D. Jacobs, R. Ramamoorthi. "Using Specularities for Recognition," *ICCV*, pp. 1512-1519, 2003.

[19] R. Schapire. "The Boosting Approach to Machine Learning, An Overview," *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.

[20] M. Singh, X. Huang. "Computing Layered Surface Representations: An Algorithm for Detecting and Seperating Transparent Overlays," *CVPR*, volume 2, pp. 11-18, 2003.

[21] J. Solem, H. Aanaes, A. Heyden. "PDE Based Shape from Specularities," *LNCS*, pp. 401-415, 2003.

[22] R. Szeliski, S. Avidan, P. Aniandan. "Layer Extraction from Multiple Images Containing Reflections and Transparency," *CVPR*, volume 1, page 1246, 2000.

[23] S. Yu. "Segmentation Using Multiscale Cues," *CVPR*, volume 1, pp. 247-254, 2004.

Figure 7: *Test images and output from the described system.* **Left:** *Six sample test images,* **Middle left:** *Edges from edge detector,* **Middle right:** *Edges that have been labeled as glass.* **Right:** *Regions of glass found by shrinking a snake around the labeled glass edges.*