Transparent Object Recognition Using Gradient Grids

Jiaqi Guo
Stanford University
jiaqiguo@stanford.edu

Abstract

Most object detection methods existing today are not tailored for transparent objects, for which local features vary as the illumination, or the background behind the transparent object changes. In addition, transparent objects have no intrinsic textures or features of their own, making it difficult to use patch descriptors. Thus, this project investigates the efficiency and effectiveness of using a template matching method based on gradient grids, which proved successful for texture-less objects, to detect transparent objects in various types of backgrounds. We first form the template using only the gradients of the object on a texture-less background, then use a similarity measure as we scan over the test image to determine if the object(s) of interest are in the image. There is no assumption that we know anything about the background in the test image. Quantization and other methods are used to optimize the method, to make real-time object detection possible, which would prove important for applications such as robotics.

1. Introduction

Object detection, especially real-time object recognition has been one of the most difficult challenges in computer vision. Because many computer vision applications require systems to adapt to new environments and recognize new objects, it is imperative to find solutions that are both robust and computationally efficient.

There are two major classes of object recognition algorithms, namely template matching and statistical methods. Statistical methods [4-5] typically require a massive amount of training data and are computationally expensive, so they are less than ideal for real-time object recognition. Template matching [6-7], on the other hand, offers some real-time solutions for most classes of objects. In particular, for highly textured objects, there are patch descriptors that can be computed efficiently for training images, and subsequently these could easily be compared with similar descriptors in test images.

However, these methods fail for texture-less objects, because not many meaningful descriptors can be extracted from texture-less objects, and in many cases these descriptors will be dominated by descriptors from a cluttered background, making it impossible to detect the texture-less object. An adapted template matching method has been used in [1-2] to successfully detect texture-less objects in cluttered backgrounds and has been shown to be faster and most robust than methods such as HoG and DOT. The reason for its success is that it quantified the image characteristic that texture-less objects still retain – their contours.

Even more challenges are posed when we consider transparent objects. They are so prevalent in our daily lives, yet little research has gone into detecting transparent objects via visual cues. Transparent objects first have no intrinsic textures of their own, rather like texture-less objects. Any “texture” that we observe is due to illumination, shadows and refraction of light through the transparent medium. We could thus use a similar template matching method to [1-2], using only gradient grids, for detecting transparent objects. But transparent objects have local features that vary as the background illumination changes or when there are other objects surrounding it, as illustrated in Figure 1. We would like to test how robust a template matching method based on object image gradients is when detecting transparent objects.

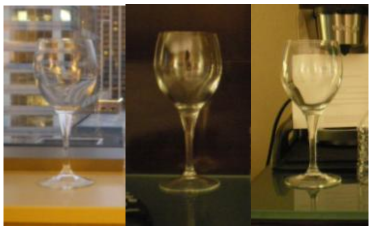


Figure 1. Demonstration of how local features of transparent objects change under different backgrounds. If we were to use simple patch descriptors for these objects, the detection results would fluctuate and be very sensitive to background changes.

In the rest of the paper, we first detail the problem and discuss related work, describe the technical approach, then examine the results of detection for various test images.

2. Related Work

Both statistical learning and template matching methods have been used for various types of object recognition, and both have their own advantages and disadvantages. Statistical learning methods typically require a large set of training images and a long training time, because their goal is to identify the general category of an object in the test image, rather than detect if an object matches something that the system has seen before. Thus, usually statistical learning methods are unsuitable for real-time tracking and detection tasks.

Statistical learning methods include Histogram of Gradients [], which summarizes the distribution of intensities within an image patch. This method has a high recognition accuracy rate, but is computationally very expensive. An even more expensive method is to use SIFT descriptors and then a learning algorithm such as Support Vector Machines to detect objects in an image [].

Template matching, on the other hand, has always had an important role in real-time object detection (such as in robotics), due to its simplicity of concept and ability to manage all different types of objects. One of its largest benefits is that it does not require a large set of training images and thus the amount of time required to train the system could be much reduced.

The crux of any template matching method lies in the similarity measure for determining the match between a training template and a test image. One of the first template matching methods involved computing the Hausdorff distance [7], which is the maximum distance from all edge points in the test image to the nearest edge point in the template, and vice versa.

Although the naive Hausdorff distance is extremely sensitive to occlusions and background noise, a workaround could be achieved by taking the maximum of only a specified fraction of distances. This removes the infinite distances between an edge point on a template and its corresponding occluded edge point in the image. However, this means that it will be necessary to estimate the maximum amount of clutter in the test image.

A variation of Hausdorff distance is the Chamfer distance between edge points in the template and the test image as the similarity measure [8]. This distance can be computed quickly using the Distance Transform of an image, but is still very sensitive to outlier edge points both in the template and test images. Regardless, the computational load is not light, and both these distances rely on finding the edge points using some form of edge detector, for example the Canny edge detector. Setting thresholds for edge detectors is always an art, and the edges detected are sensitive to variation in illumination and background clutter, so both these methods are not completely robust.

Instead of using image edge points, there are also methods that use image gradients and the similarity measure is then defined as the dot product between the template edge gradients and the test image edge gradients. However, these require dense sampling for accurate results, and are usually not computationally practical. Moreover, one must be sure to normalize the products in such cases or there will be false positives due to larger gradients in the background clutter.

The method used in this project overcomes the dense sampling required, and also does not require a large amount of training data. Each transparent object is represented with a set of templates from various viewpoints, and each template is a patch of dominant image gradients. The templates are then used to detect the objects of interest in test images.

3. Technical Approach

3.1. Data Collection

We want to use efficient template matching to detect transparent objects. The training data collected to form our model templates is a set of two-dimensional images of transparent reference objects against a clean and uncluttered background. One dataset had objects captured from multiple viewpoints (see Figure 2), to test if the object detection is indeed invariant to viewpoint and slight distortion. The second dataset had one template (viewed from the front) for each object, the main objective of which is to test how robust the method was to local feature changes because of illumination or background changes. For each training image, a binary mask was created for each of these images to facilitate creating a model template that is free of background clutter.

These data will then be processed to form the set of model templates, which is the basis of our detection system. These templates will be matched against patches on test images of numerous levels of difficulty: images free of background clutter, images with some background clutter, and images with a lot of background clutter.

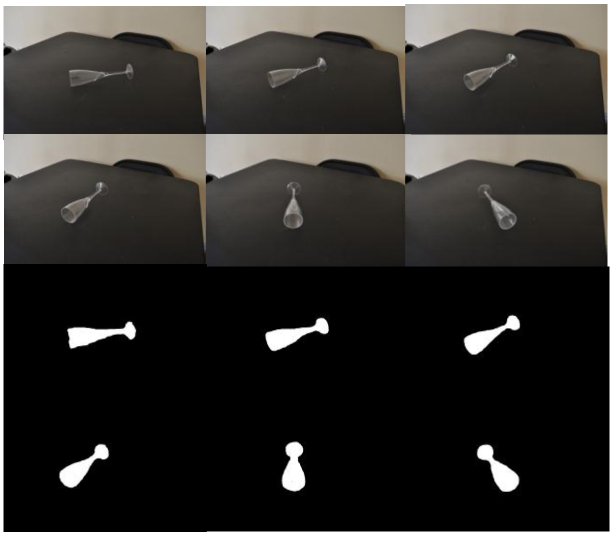


Figure 2. This is an example of a set of training reference images and binary masks for one transparent object, from multiple viewpoints.

3.2. Summary of Algorithm

ALGORITHM 1

```
Set T = 8
// Training step
For each object to detect
    For each viewpoint
        // compute the model template
        For each location in the object's bounding box
            Compute the gradient orientation
            Quantize each gradient orientation
        For each location in template // second pass
            Set gradient orientation at this location to be the quantized orientation that occurs the most often in its 3x3 neighborhood // smoothing
            Spread quantized orientation at this location to

 // Pre-compute response maps
        For each possible combination of quantized gradient orientation
            Calculate the maximum response of that combination to a single quantized gradient

// Testing step
For each test image
    For each location in image
        Compute the gradient orientation, and smooth
    For each learned template
        For each template-sized patch in image
            Calculate the similarity score
            if similarity score > threshold
                annotate this patch as an object instance of the template it's currently being compared to
```

This template matching algorithm allows not only the object class to be identified, but also provides a rough pose estimation given that for each object, the templates learnt cover the range of poses that are of interest for that object. For example, for the transparent glass in Figure 2, if we were to just use those six templates, it would give us a range of roughly 90 degrees on the horizontal plane. If more poses are to be detected, more templates should also be added.

3.3. Model Template Calculation

Gradient orientations are mostly invariant to illumination and background changes, which are problems with all binary edge detectors. Furthermore, because with transparent objects, local features within the contours of the object may change as backgrounds and illumination change, so image gradients that outline the contour of the object are more reliable than other object descriptors.

Thus, we would like to exploit only the gradient orientations of the object. The orientation of the gradient is calculated for each point in the image. To completely remove the problems of a binary edge detector, the gradients are then normalized.

To further make the method robust, we take the maximum of the gradients in the red, blue and green channels so that color or illumination variation do not result in any bias in gradient orientations. Thus, for an image ***I*** with R, G, B color channels, the gradient orientation ***G(x)*** at location *x* in an image G is defined as

$$G(x) = ori\left(\hat{C}(x)\right) \qquad (1)$$

where

$$\hat{C}(x) = \ argmax_{C\in\{R,G,B\}}\left|\left|\nabla_x C\right|\right| \qquad (2)$$

To handle false-positive gradients that result from noise, gradients which have norms below a certain threshold are ignored, to eliminate some weak gradients due to noise in the image.

3.4. Optimization of Template Computation

To improve speed and robustness, the gradients are then quantized into 9 distinct bins based on their angles. Also, because we are interested in characterizing the contour of the transparent object, we do not differentiate between gradient orientations in completely opposite directions, and limit our gradient orientations to 0 to 180 degrees only (see Figure 3). To further eliminate gradients that are a result of noise, we take the gradient orientation at a location to be the quantized gradient orientation that occurs most often (dominant gradient orientation) in the 3 by 3 neighborhood of the location.

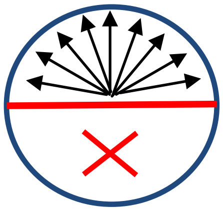
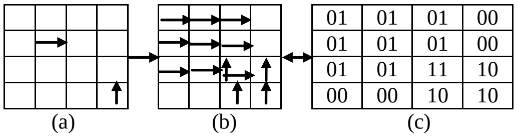


Figure 3. Gradient orientations are quantized into 9 bins as shown above, with no orientations in the range 180 to 360 degrees. It is possible to quantize to a different number of bins, but 9 bins seems precise enough for our purposes.

We then "spread" the gradients by adding the gradient of one individual pixel to be the gradients in a T x T neighborhood of the pixel (see Figure 4). This allows us to speed up the detection step after we have our templates, and also allows for slight distortion or misalignment when detecting the object in the test image.



01	01	01	00
01	01	01	00
01	01	11	10
00	00	10	10

Figure 4. Spreading the gradient orientation around for robustness.
(a) In the first grid, the arrows represent the original gradient orientations obtained from the training image.
(b) Spreading the gradient orientation at each location to a 3 x 3 neighborhood results in the second grid of orientations. If we compare this template of orientations to a patch of orientations in a test image, a slight distortion in the test image could still result in a match.
(c) The third grid shows how the quantized orientations are actually stored. If we were only quantizing to two bins, then the storage grid would look like this, with the second bit turned on if some location has a horizontal orientation component, and with the first bit turned on if some location has a vertical orientation component.

To increase efficiency, after quantization, gradient orientations are stored using a binary string per location, as demonstrated in Figure 4. Each '1' in the binary string represents that that pixel has a component in the corresponding gradient orientation.

A visualization of the contours after gradient orientations have been computed, dominant gradients found and quantized, and then spread, is presented in Figure 5. It is obvious that the contours are extremely clear, and because they are normalized, there is no worry for false positives resulting from overly dominant gradients due to illumination or background clutter.

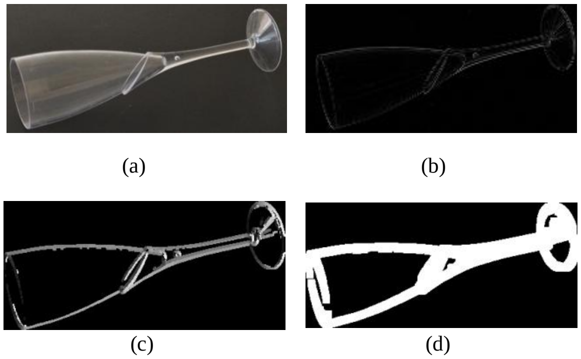


Figure 5. (a) Original training image
(b) Gradient image computed using the normal approach
(c) Gradient image after quantization. The gray levels each represent one quantized orientation.
(d) Gradient image after quantization and spreading

3.5. Similarity Measure

The similarity measure aims to be robust to both background clutter, and small translations and deformation. Given a gradient orientation template $\boldsymbol{T}$ and test image $\boldsymbol{I}$, we define the similarity score as

$$E(I,T,c) = \sum_{r\in Pos} max_{t\in R(c+r)} \left|\cos\left(ori(T,r) - ori(I,t)\right)\right| \qquad (3)$$

Here, *Pos* is the set of positions within a template T which have non-zero gradient orientations, so we do not have to check all positions in a given template, since many pixels on a near-texture-less object would unavoidably have no gradient orientations at all. Note also at each location of the image, we are taking the maximum of the match in orientation with the template in a neighborhood of the location, so that the method is even more robust to slight distortions in the test image.

Having to take the difference of the orientation, calculate the cosine of that, and find the maximum of that value in the neighborhood of every single location in the test image and for every single template is extremely time-consuming. It is possible to speed up this computation by pre-computing the "similarity response" of each possible combination of quantized gradients (in template images) to each possible quantized gradient orientation (in test images). Thus, the similarity score can be obtained by a few lookups in a hash table of similarity responses.

To determine if we have detected an object, we impose a threshold on the similarity score, and only patches within the test image with higher scores than could be potential instances of the object in interest. For this project, a few threshold values were tried and the one that gave the best true positives versus false positives ratio was kept.

There is one more optimization in terms of speed that can be done. Because we had previously spread our gradient orientations in the templates, it is now possible to not have to scan through patches one pixel by one pixel, which is a very time consuming process. We can skip T pixels at a time, and would not miss any important information.

4. Results and Discussion

4.1. Small displacements and distortions

This method proved very robust to small displacements and distortions. Rotating the transparent glass on a texture-less table step by step, a set of 60 images were collected for the glass, that varied from 0 degrees all the way to 360 degrees on the horizontal plane. Of these 60, 15 were taken as training images to form the model templates, and the rest were treated as testing images. Each image in this set was in a slightly different orientation, and may also have been slightly displaced from each other because the rotation was manual and imperfect.

The above described method picked up almost all of these objects in the image, regardless of the small shift in translation or rotation. Tests were done with other transparent objects such as cups and water bottles on similar texture-less backgrounds, and Table 1 summarizes the effectiveness of the method on these objects on clean backgrounds.

Object	True Positives	False Positives
Glass	100%	0%
Water Bottle	100%	0%
Drinking cup	99%	1%

With these results, we can conclude that on uncluttered backgrounds, the template matching method can effectively detect objects that are slightly distorted, as long as the background remains clean and uncluttered, like in the training images.

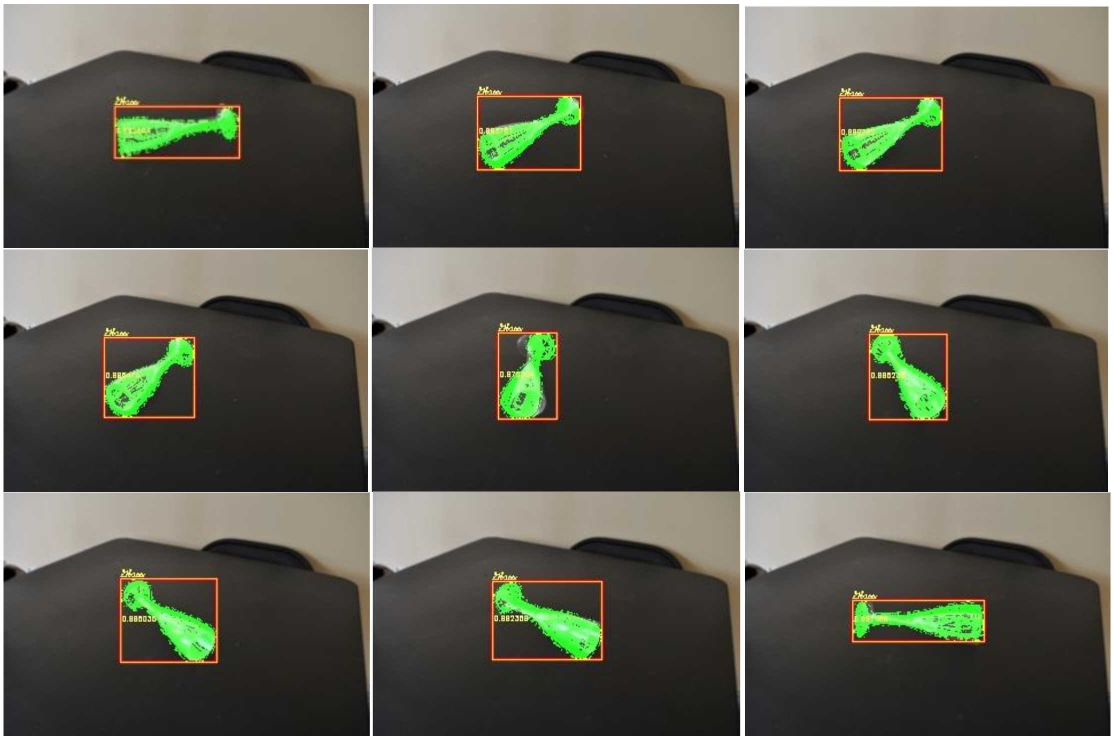


Figure 6. Detection of the transparent glass in various orientations and small displacements. The glass in each case was found successfully, although the original training templates were not in the exact same (but close) orientation and position as these glasses.

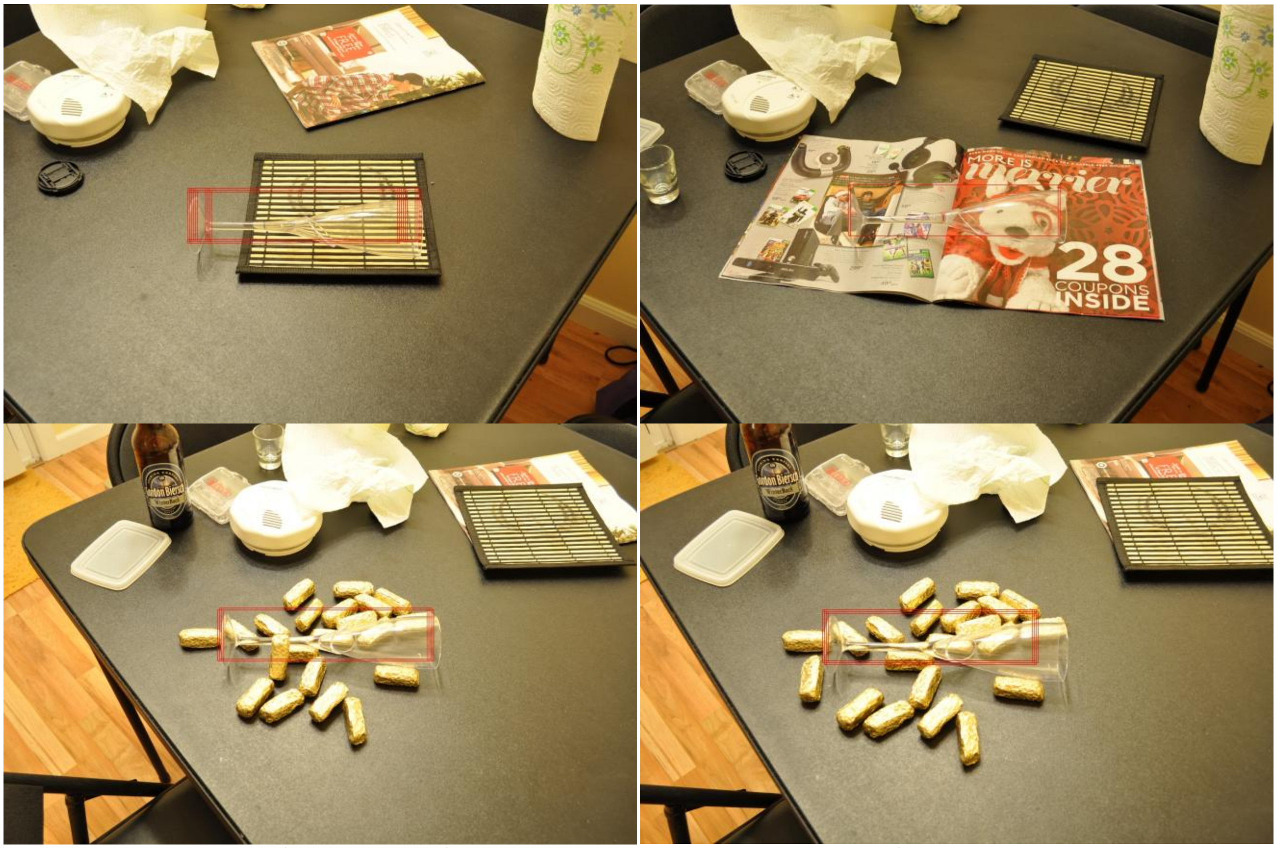


Figure 7. Even in the presence of various degrees of background clutter, template matching managed to find the transparent glass. In the last two images, the scales of the glasses are different, but both objects were detected successfully, suggesting that this method is robust to slight changes in scale as well.

4.2. Background clutter

In cluttered backgrounds, we would expect to see more false positives, since there is a much increased probability of something in the background having a similar contour to the transparent object of interest. Indeed, if I lower the threshold score to 0.5, I see many positives being detected around the glass itself. However, picking a score threshold of 0.9, the glass was detected accurately in all the images shown above with any false positives, and the detection rate was also high in the other test situations. The true positives to false positives rate was 98% - 2%, which is on par with some of the better statistical learning methods and better than some template matching methods. False positives mostly showed up when there was something similar in the background to the cup, such as the mat with horizontal lines (because the cup orientation is pretty much horizontal, the mat looked similar to the detector, although it did have a lower similarity score than the actual cup).

In the third and fourth image above, the images are almost similar but the glass is slightly bigger in the fourth image than the second image, but was still successfully detected. We can see from this that the template matching method is also robust to small changes in scale of the object in the test image. Amongst the images that were tested, the range of scales, compared to the training image object scale, which were successfully detected by the system, was (1.0, 1.5).

4.3. Notes on similarity score

There are many ways to set the threshold on the similarity score for determining if some image patch is indeed some meaningful object. We could just take the absolute value of the similarity score and give it a threshold. However, this number would always vary as we test with different objects, since the number of gradient pixels on each object would be different, and would vary with distance to the object and the object’s orientation. Thus, this is not a robust scoring method.

One could also use the number of gradient points in the template image before or after spreading the gradient orientations, but false positives that happen to have just a few more local features could score higher than the actual object that has less local features, because of the spreading.

Thus the solution taken in this project was to increase the neighborhood range when smoothing gradient orientations in the test images. Testing various sizes of neighborhoods to smooth over, the best size happened to be T, the size of the neighborhood we chose to spread the orientations over, which was 8 in this case. Normalizing the similarity score by the number of gradient pixels in the template after smoothing, a threshold of 0.9 was able to give me the high detection rates and low false positive rates.

5. Future Work

Depth information is a new modality that can be made use of, especially with the now prevalent Kinect sensors that provide valuable 3D depth information. This information could help make detection results more robust, especially because depth sensors usually show impossible depth at the location of a transparent object. Coupling this information with the gradient orientation similarity measure may improve the detection results.

Furthermore, it would be interesting to try to model the refraction of light through transparent objects. Illumination does indeed greatly affect the local features of especially glass-made transparent objects, and if it were possible to accurate model the light refraction through the glass or somehow account for it, it would be possible to extract the real contour of the object instead of having to worry about local features.

On the same note, because illumination affects transparent (and refractive) objects so much, it would be of interest to study the detection of objects not with one test image, but with two images under different lighting conditions – perhaps one that is taken with flash and one without. The change in light refraction could help the system detect what is constant in the equation, and could help to improve detection results.

6. Conclusion

The template matching method based on gradient grids works well for transparent object detection, and achieves a high detection rate that can be compared to the detection rate in statistical learning and other template matching methods for general (non-transparent) objects. It would be of interest to study the characteristics of transparent objects further, and make use of that knowledge to improve the system's robustness to noise and clutter, and perhaps even to object scale.

Acknowledgements

The author(s) of this report would like to thank Dr Fei-Fei Li and the TAs for their help and guidance through the course.

Future Distribution Permission

The author(s) of this report does not give permission for this document to be distributed to Stanford-affiliated students taking future courses.

References

[1] S. Hinterstoisser et al. Gradient Response Maps for Real-time Detection of Texture-less Objects. In PAMI 2011.

[2] S. Hinterstoisser et al. Multimodal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes. In ICCV 2011.

[3] M. Fritz et al. An Additive Latent Feature Model for Transparent Object Recognition. In NIPS 2009.

[4] C. Huang et al. Vector Boosting for Rotation Invariant Multi-View Face Detection. In CVPR 2005.

[5] P. Viola and M. Jones. Fast Multi-view Face Detection. In CVPR 2003.

[6] D. Gavrila and V. Philomin. Real-Time Object Detection for "Smart" Vehicles. In ICCV 1999.

[7] D. Huttenlocher et al. Comparing Images Using the Hausdorff Distance. In TPAMI 1999.

[8] G. Borgefors. Hierarchical Chamfer Matching: a Parametric Edge Matching Algorithm. In IEEE Transactions on Pattern Analysis and Machine Intelligence 1988.