Enforcing 3D Constraints To Improve Object and Scene Recognition

Robin Hewitt
Hewitt Consulting, San Diego, CA, US
rhewitt@acm.org

Luis Goncalves
Evolution Robotics Retail, Pasadena, CA, US
luis@evoretail.com

Mario Enrique Munich
Evolution Robotics, Pasadena, CA, US
mario@evolution.com

Abstract

This paper presents an extension to David Lowe's well-known object recognition algorithm based on his Scale and Feature Invariant Transform (SIFT). One of the benefits of Lowe's SIFT-based method is that it can recognize objects from only three keypoints. While this capability can be useful in circumstances where the cost of a false negative is high, it's often the case that false positive are an equal, or greater, concern.

We extend Lowe's algorithm by adding the ability to use 3D constraints during matching. These constraints essentially eliminate false positive matches. We combine our extension with the original algorithm to retain the recognition power of the original method while adding significant robustness against false positives, thereby increasing overall classification power. In addition to improving recognition, our extension returns 3D pose information. Yet, it adds very little computational overhead to Lowe's original algorithm.

1 Introduction

Advances to object recognition during the past decade have given rise to new algorithms. One of the most successful of these is the algorithm developed by David Lowe, based on the Scale Invariant Feature Transform (SIFT) [6]. SIFT (or derivatives of it) has been integrated into a number of commercial products, including Sony's Aibo, Bandai's NetTansor robots, and the visual Simultaneous Localization and Mapping (vSLAM) system [4] by Evolution Robotics.

Lowe's method has many attractive qualities. It recognizes multiple objects in query images based on minimal training input. It's simple to use: no specialized expertise, dataset, nor equipment are required to train new models. Discrimination between multiple

learned objects is handled efficiently. It's completely robust to changes in scale and to in-plane rotations, and it accommodates mild perspective distortions that arise from out-of-plane rotations.

Where SIFT has difficulty, is in handling distortions due to strong 3D perspective effects. This shortcoming limits its usefulness in applications that require robustness to these distortions. Strong 3D effects create several problems for SIFT.

The first is a change to keypoint descriptors. SIFT descriptors model local appearance as a histogram of gradient direction. Strong perspective effects warp image regions, changing the descriptors, thereby decreasing the similarity between descriptors in training and query views. Mikolajczyk and Schmid [8] have introduced an affine invariant descriptor that removes the problem. It has been used successfully in a number of applications, including recent methods for 3D object recognition ([1], [11].

Other perspective difficulties for SIFT arise from parallax effects, however. Lowe uses an affine transformation to approximate mild perspective effects. For objects that are essentially flat or smoothly convex, this is a reasonable approximation. For objects with limblike protrusions, however, parallax effects can be important, and these are not well approximated by an affine warp. Figure 2 illustrates this. As the toy bear rotates about its body's centerline (2b and 2d), head and torso remain nearly stationary within the image, but arms and legs are displaced much farther. Other objects that exhibit a similar parallax effect include airplanes, chairs, and tables.

Parallax is also a significant factor in visual SLAM applications. Figure 4 illustrates this. As the camera moves through the furnished living room, keypoints generated by the chair in the foreground move farther between views than do keypoints generated by the more distant fireplace.

The third problem is that, when out-of-plane rotation is large, an affine warp may be a poor approximation, even when the object is flat or convex. This is also a parallax effect. It's most apparent when a large portion of the object's surface is approximately planar (because self occlusion is then low), and the perspective effect is strong.

In this paper, we present an extension to Lowe's algorithm that accommodates these effects. Our extension consists of two changes to the final step in Lowe's algorithm. The first is extremely simple. Instead of approximating perspective effects by an affine transform, we fit a homography. This small change, by itself, is enough to significantly improve recognition scores on relatively flat objects when out-of-plane rotation is large.

Our second change is more substantial. We apply a 3D constraint based on finding two homographies from different bins in Lowe's Hough transform. Two homographies not only constrain 3D pose information, they overconstrain it. This overconstraint on 3D geometry allows us to apply a very reliable error measure based on self consistency. We show how this self-consistency measure can be used to improve overall recognition performance.

These extensions increase SIFT's discrimination power when parallax effects are present with little additional computation overhead. Further, the same method seamlessly accommodates both planar objects and objects with more complex 3D shape.

Since we retain the machinery of SIFT, the choice to apply our extension is easily toggled on and off. It can be applied whenever false positives are a particular concern, and omitted to enable more lenient recognition. When the extension is enabled, it provides 3D pose information in addition to recognition results.

2 Related Work

Early work on 3D recognition used a visual manifold, with 3D objects or scenes represented as images from several viewpoints [9]. Recently, new methods ([1], [11], and [5]) have begun to emerge that explicitly address the problem of 3D object recognition in the presence of strong perspective effects.

In [1], Ferrari et al. use a combination of local features and region growing to model spatial relationships while providing robustness to heavy occlusion and clutter. However, viewpoint invariance still relies on exhaustive comparisons with every training image. Kushal and Ponce [5] extend Ferrari's work. They build an explicit 3D model of each object from stereo views taken at seven to twelve vantage points in a ring around each object. However, the specialized equipment (stereo rig and turntable) needed for training make this approach unsuitable for some applications.

In [7], Lowe presents an extension to his own SIFT-based recognition method in which features from multiple views of an object are linked. He proposes that an alternative extension would be to solve explicitly for 3D structure during matching. This is the approach we take in this paper. Unfortunately, only qualitative testing was done in [7], making it difficult to compare these two approaches.

Our extension to Lowe's method uses planar homographies. Homographies have been used extensively in object recognition methods ([2], [12]). However, the existing homography-based methods that we're aware of rely on locating planar surfaces in an object. Our approach differs, in that we don't require that homographies correspond to planar surfaces. Consequently, we don't rely on being able to locate planar surfaces, nor even on these being present in the image.

3 Methods

Below, in 3.1, we briefly outline Lowe's SIFT-based recognition method. In 3.2, we describe our extension for handling parallax effects. Last, in 3.3, we describe how we combine our 3D extension with Lowe's algorithm to improve recognition performance.

3.1 A Brief Overview of Lowe's SIFT-Based Recognition Method

Lowe's object recognition method [6] is based on correspondences of salient points at multiple image scales. Each salient point is represented by its SIFT descriptor – a circular histogram of gradient directions.

To compare training and query images, K nearest neighbor matching is used to find putative correspondences between keypoints from all training images and keypoints in the query image. Each keypoint-to-keypoint correspondence casts one vote for a combination of training image, pose, and scale. Votes are accumulated in a Hough transform. The Hough bins that receive the most votes constitute candidate hypotheses for the presence of an object at a particular location, scale, and in-plane rotation.

Last, Lowe validates candidate hypotheses by iteratively applying a least-squares fit for an affine transformation to the keypoints in a Hough bin. Outlier keypoint matches are removed at each iteration. Hypotheses that are reduced to fewer than three keypoint matches are rejected as incorrect.

3.2 Our 3D Extension

3.2.1 From Similarity To Homography

Hough transforms are typically used to detect shapes by accumulating votes from widely separated pixel locations. In Lowe's method, however, the Hough transform plays a different role. Instead of accumulating votes for structures within an image, it accumulates votes for Similarity transforms that map keypoints from one image to another. Lowe's Hough transform quantizes 4D Similarity space (t_x, t_y, s, θ). The votes in each Hough bin represent statistical support for the Similarity transform

$$\mathbf{S}_b = \begin{bmatrix} \bar{s}_b \cos\bar{\theta}_b & -\bar{s}_b \sin\bar{\theta}_b & \bar{t}_{xb} \\ \bar{s}_b \sin\bar{\theta}_b & \bar{s}_b \cos\bar{\theta}_b & \bar{t}_{yb} \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where $(\bar{t}_{xb}, \bar{t}_{yb}, \bar{s}_b, \bar{\theta}_b)$ give the translation, scale, and orientation coordinates at the center of bin b. In the last step of his recognition method, Lowe upgrades each $\mathbf{S}_b$ from a Similarity to an Affinity.

Our first extension to Lowe's algorithm is very simple. We upgrade to a Homography, rather than to an Affinity. The Homography upgrade could be done in various ways – with a least-squares fit, with RANSAC, and so on. The method we found worked best is to use gradient descent to convert the Similarity into a Homography by minimizing an error term, $f(e)$. The e here is a vector of reprojection errors for each point pair:

$$e = x_2 - \mathbf{H}x_1, \tag{2}$$

where, x_1 and x_2 are the keypoint locations in the training and test images, respectively, and $\mathbf{H}$ is the homography transform. The function $f(e)$ is a sinusoidal damping function that minimizes the effect of outliers at each gradient-descent iteration. The damping function we use is

$$f(e_i) = sgn(e_i) * ln(1 + |e_i|). \tag{3}$$

At each iteration, we discard outliers and add keypoint matches that were not part of the initial Similarity, but which are inliers to the Homography in both location and scale. This process converges very quickly. Typically, two iterations are enough.

When perspective effects are strong, a homography is often a good approximation for objects with a dominant planar surface, and even for many objects that are not planar. Figure 1 demonstrates the improvement from this simple change.

If a large percentage of the training view's keypoints are matched as inliers, a positive result is returned for the training image. If no valid homography is found, we reject this Hough bin. If, however, a homography is found that has some support, but there are many keypoints in the training image that aren't inliers, we continue to the next step – applying a full 3D constraint. This process is described in the next subsection.

3.2.2 Using Two Homographies To Define 3D Pose

For a 3D fit, we could use RANSAC to search for two additional keypoint matches that are not on the homography, compute the fundamental matrix from these points plus the homography, and evaluate the match based on inlier support. In fact, in our first attempt

to enforce 3D constraints for object recognition, we did exactly this. The difficulty here, however, is that the fundamental matrix by itself is a very weak constraint. It only enforces a point to line mapping. In cluttered query images (such as those in Figure 6), many accidental keypoint matches may conspire to support an incorrect epipolar geometry – or, worse, an incorrect training image. Conversely, the correct solution may have relatively few supporting keypoint matches, especially in the presence of heavy occlusion.

Instead, we make use of the fact that we already have Hough bins, sorted by votes. Using the same method as for $\mathbf{H_1}$, we look at one or more additional bins to see if one of them yields a second homography, $\mathbf{H_2}$. To avoid rediscovering $\mathbf{H_1}$, we mask off an epsilon region around each of its inlier keypoints.

From [3], given two planar homographies, $\mathbf{H}_1$ and $\mathbf{H}_2$, the fundamental matrix, $\mathbf{F}$, can be computed as

$$\mathbf{F} = [\mathbf{H}_i\mathbf{e}]_\times \mathbf{H}_i, \tag{4}$$

where $i = 1$ or 2, and $\mathbf{e}$ is the non-degenerate eigenvector of ${\mathbf{H}_2}^{-1}\mathbf{H}_1$. (Here, and below, $[\mathbf{a}]_\times$ represents the 3×3 skew-symmetric matrix s.t. $[\mathbf{a}]_\times \mathbf{x} = \mathbf{a} \times \mathbf{x}$.)

The significant benefit we gain by computing $\mathbf{F}$ in this way is that we can then compute a direct measure of 3D self-consistency error that's *independent* of inlier support to measure goodness of a match. First, we find solutions to Equation 4 for $i = 1$ and 2:

$$\mathbf{F}_1 = [\mathbf{H}_2\mathbf{e}]_\times \mathbf{H}_2, \text{ and } \mathbf{F}_2 = [\mathbf{H}_1\mathbf{e}]_\times \mathbf{H}_1. \tag{5}$$

We then make use of the constraint that a homography $\mathbf{H}$ is compatible with a fundamental matrix $\mathbf{F}$ if and only if the matrix $\mathbf{H}^T\mathbf{F}$ is skew symmetric [3]. In other words,

$$\mathbf{H}^T\mathbf{F} + \mathbf{F}^T\mathbf{H} = 0. \tag{6}$$

We calculate self-consistency error, e, as

$$e = \max\left(\left|{\mathbf{H}_1}^T\mathbf{F}_1 + {\mathbf{F}_1}^T\mathbf{H}_1\right|, \left|{\mathbf{H}_2}^T\mathbf{F}_2 + {\mathbf{F}_2}^T\mathbf{H}_2\right|\right). \tag{7}$$

3.2.3 Virtual Surfaces

Using Lowe's Hough bins, valid, plane-induced homographies can be identified between views of objects that are not themselves planar. Figure 2 shows keypoint matches for two homographies induced via sets of point matches associated with planes that intersect a toy bear. Each homography was initialized by the keypoint matches in a separate similarity bin. In the same way that a Hough transform for lines finds distributed, but collinear, segments in an image, the Hough transform in Lowe's algorithm helps bring out homographies from "virtual surfaces" that may have distributed, rather than localized, support.

3.3 Using 3D Constraints To Improve Recognition

As test results below show, adding our 3D constraint to Lowe's method essentially eliminates false positive matches. This characteristic makes it a useful complement to the original method, in which false positives can be a problem. The question remains, then, how to smoothly integrate our extension with Lowe's original algorithm to get the best overall performance.

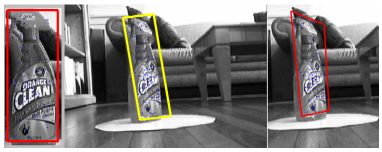
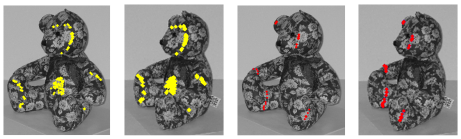


Figure 1: Simply replacing an affine fit (yellow) with a homography fit (red) improves recognition when out-of-plane rotation is high.



(a) $\mathbf{H_1}$ (b) $\mathbf{H_1}$ (c) $\mathbf{H_2}$ (d) $\mathbf{H_2}$

Figure 2: Homographies $\mathbf{H_1}$ and $\mathbf{H_2}$ need not correspond to object surfaces. Here, our method found two virtual planes that transect a rounded toy bear.

Provost and Fawcett [10] provide a principled answer to this question. They show that, when ROC data are available for several classifiers, classification quality can be maximized to be equivalent to the convex hull of all classifiers. The strategy for achieving this classification quality is based on the insight that the tangent line to the ROC curve can be expressed as

$$m = \frac{C_{fp} * NEG}{C_{fn} * POS}, \tag{8}$$

where,
m is the slope of a line in (FPR, TPR) space,
$\frac{NEG}{POS}$ is the ratio of true negatives to true positives,
C_{fp} is the cost for a false positive, and
C_{fn} is the cost for a false negative.

It follows that the optimal classifier in a particular use context is the one that lies on the convex hull at the point where the convex hull's tangent has the value computed by Equation 8. When the tangent touches two classifiers on the convex hull, the optimal strategy is to choose one of these two classifiers by a weighted coin toss. The weight given to each classifier is equal to its distance from the point on the convex hull that has the target FPR value.

In practice, the values on the right-hand side of Equation 8 are often unknown. In that case, the convex hull still serves as a guide for combining classifiers, and the choice for which classifier to use in a given context can be determined empirically.

4 Testing

4.1 Testing With SLAM Images

4.1.1 Experimental Setup

We evaluated SIFT matching, with and without our extension to handle parallax effects, on the task of scene matching in an indoor (home) setting. Training images consisted of six partially overlapping views, shown in Figure 3. Test images consisted of 454 views of the the same environment. 358 of these overlapped with one or more training views, and 97 contained views of areas that were not covered by the training images. The images

Figure 3: Training images for the visual SLAM experiment.

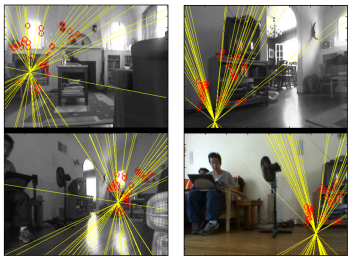
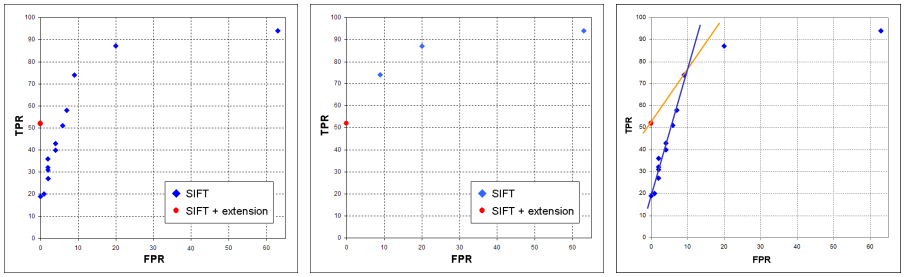


Figure 4: Example scene matches.

consist of video frames from a webcam mounted on a robot. The robot was driven, by remote control, through a house. The training views in these tests consist of frames taken near the beginning of the sequence.



(a) All ROC points (b) The convex hull (c) Change in ROC tangent

Figure 5: The ROC data for scene recognition.

4.1.2 Results

To analyze results from the SLAM tests, we pose a binary classification problem. The two classes are "known" and "unknown" location. The use of several, potentially overlapping, images to represent the "known location" class is then analogous to representing a multimodal distribution with one training sample per mode. Figure 5 shows ROC data for classification with and without 3D constraints. As this figure shows, at FPR=0, the ROC point for recognition with our extension lies well above the ROC point for Lowe's original method (TPR=52 with extension versus TPR=19 without).

From Equation 8, when $\frac{C_{fp}*NEG}{C_{fn}*POS} \geq (\frac{74-52}{9-0} = 2.4)$, our extension is the best classifier. Only when this ratio drops below $\frac{87-74}{20-9} = 1.2$ does Lowe's original method become the better choice. Between 2.4 and 1.2, choosing between these methods by a weighted coin toss is a better strategy than relying exclusively on either one.

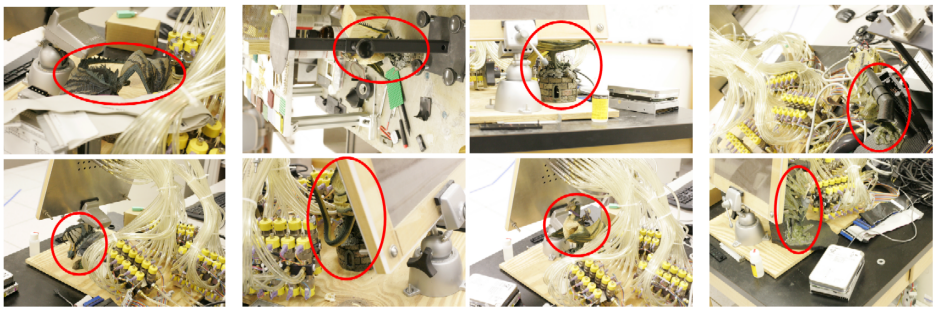
In addition to scene matching, using our extension gives 3D information. Figure 4 shows the epipolar geometry for example scene matches.

4.2 3D Object Recognition

4.2.1 Experimental Setup

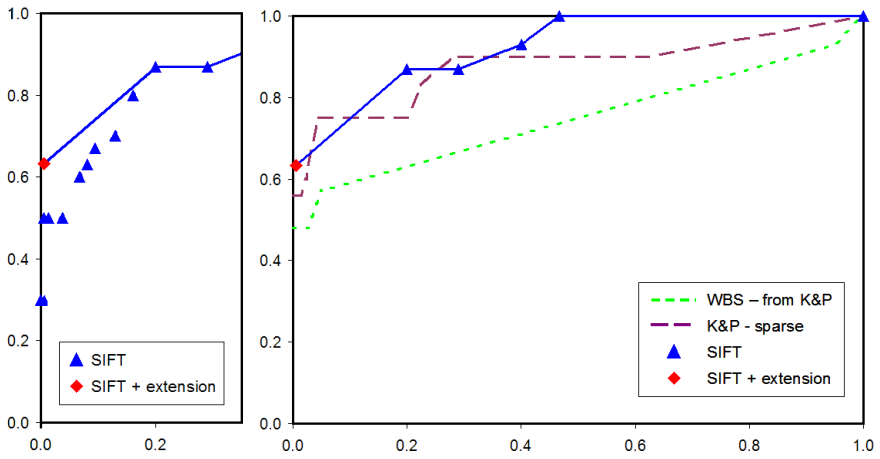
We also tested our extension to Lowe's method on 3D object recognition using the "geometrically complex" objects (two dragons and the chest buster) from the dataset that Kushal and Ponce used to evaluate their 3D recognition method [5]. The test images in this dataset are very challenging, consisting of 3D objects in various poses amidst dense clutter. In many, the objects are more than half obscured by feature-rich clutter.

Kushal and Ponce make use of 14-24 training images (seven-twelve stereo pairs) for each object. Although we could also have used all training images, we chose instead to use a small subset – less than 1/5 of their training images – since that better reflects how we expect our method to be used. We tested against all 80 test images.



(a) Small dragon (b) Green dragon (c) Chest buster

Figure 6: Examples of True Positives for 3D object recognition with our extension.



(a) Convex hull (b) Comparisons

Figure 7: ROC data for 3D object recognition tests.

4.2.2 Results

Figure 7 shows the ROC curve for the 3D object recognition tests. As 7a shows, at FPR=0, the ROC point for recognition with our extension again lies well above the ROC point for Lowe's original method (TPR=63 with extension versus TPR=30 without).

In [5], Kushal and Ponce follow a sparse, feature-based matching step with a dense, region-growing step to achieve very good results. Ours is a sparse method, which could also be followed by region growing. It's therefore interesting to compare our sparse method with theirs. They used DoG and Harris corner features, color histograms, and affine-rectified SIFT descriptors. We used only Lowe's original SIFT descriptors, and did not rely on color cues. As Figure 7b shows, recognition rates are similar, even though we used fewer than one fifth the number of training images. Both sparse methods noticeably outperform wide baseline stereo matching.

4.3 Consumer Products Dataset

4.3.1 Experimental Setup

Finally, we tested our extension to SIFT on Evolution Robotics Retail's consumer-products dataset. This dataset contains 255 training views of packaged consumer products, and 136 test images. Training images consist of up to six views of each product. A product barcode is associated with each training image. Test images were obtained under real world conditions. Ground truth for the test images consists of both the barcode and the most similar training image. The most visually similar training view for each test image was selected manually, prior to testing.

4.3.2 Results

Table 1 compares SIFT's performance with and without our extension. Two recognition results are reported for each method: Percent Correct View Matches and Percent Correct Product Matches. For View Matching, a correct match is scored only when the training view that is matched is the one that had been manually selected as being most visually similar to the test view. For Product Matching, a match is scored as correct if test and training images are linked to the same barcode, regardless of which view was matched.

5 Summary

Our extension to Lowe's method is efficient. The first homography, $\mathbf{H_1}$, adds zero overhead, since it directly replaces Lowe's step of computing an Affine warp by iterative least squares fit. To find $\mathbf{H_2}$ (or to give up on doing so) requires examining one or more additional Hough bins. However, Lowe's original algorithm also requires fitting transformations to multiple bins. With our extension, Lowe's method can be made significantly more robust to false positives, and the extension is easily toggled on and off as needed.

Acknowledgments

This work was funded by the US Office of Naval Research, Contract N00014-06-C-0033.

Method	View Matching	Product Matching
SIFT	44	89
SIFT + extension	68	97

Table 1: Percent correct matches on the consumer product dataset.

References

[1] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *European Conference on Computer Vision, Proceedings*, pages 40–54, 2004.

[2] L. Van Gool and A. Zisserman. Grouping and invariants using planar homologies. *Workshop on Geometrical Modeling and Invariants for Computer Vision*, 1995.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[4] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vslam algorithm for robust localization and mapping. *IEEE International Conference on Robotics and Automation, Proceedings*, pages 24–29, 2005.

[5] A. Kushal and J. Ponce. Modeling 3d objects from stereo views and recognizing them in photographs. *European Conference on Computer Vision, Proceedings*, 2006.

[6] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1999.

[7] D. G. Lowe. Local feature view clustering for 3d object recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[8] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *European Conference on Computer Vision, Proceedings*, 1:128–142, 2002.

[9] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[10] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[11] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006.

[12] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16(1):57–99, 1995.