Augmenting Corner Descriptors

Paul L. Rosin

Abstract

A failing of many grey-level corner detectors is that they do not extract most of the attributes of a corner apart from its strength. This paper provides several post-processing techniques for determining additional corner attributes (i.e. colour, orientation, subtended angle, and contrast). Corner matching processes can use this additional information to resolve otherwise ambiguous correspondences and to eliminate corners whose attributes do not match certain criteria.

1 Introduction

A common paradigm in computer vision is the extraction of simple low-level features from the image followed by further processing based on these features. However, this approach accounts in part for the failure of contemporary computer vision systems to operate in a general purpose and robust manner. The simple features provide an impoverished description, making the analysis of the scene based on this inadequate data – which is usually incomplete, imprecise, and ambiguous – extremely difficult. A richer description of image features would improve computer vision systems since additional constraints enable faster convergence to a less ambiguous solution. Many image features are commonly under-utilised in computer vision [17] since much of the information present is discarded without being used.

This paper considers grey scale corners which are a popular low-level image feature used for matching in object recognition, stereo and motion analysis, and image registration. Most grey scale corner detectors assume an idealised corner that is sharply pointed and has straight, steep edges, and return just a single value measuring the "cornerity" or "strength" of the corner (e.g. [2,6–8,13,14,20,21]). However, corners rarely appear like this in the real world. Due to manufacturing limitations, wear and tear, streamlining, aesthetics, and so forth, corners are more typically rounded, blunted, blurred, ragged, textured, etc. Some of these attributes are listed and described in more detail below. In particular, cornerity is usually derived from a subset of the following properties:

- *Position* – location of the corner. For round corners the two edges of the corner can be extrapolated and the position defined by their intersection. Alternatively the position can be constrained to lie on the rounded edge.
- *Subtended Angle* – sometimes called "pointiness" or aperture. This is the angle between the corner's two edges.
- *Orientation* – the angle of the orientation of the whole corner.
- *Edge Shape* – the edges leading to the corner can be straight or curved (either concave or convex).
- *Edge Texture* – whereas the edge shape refers to the overall shape of the contour, the edge texture is the finer structure of the contour, e.g. wiggly, spiky, undulating, etc.
- *Contrast* – the difference in intensity between the corner (foreground) and the non-corner (background).
- *Edge Profile* – the manner in which the image intensity varies across the edge, e.g. step, ramp, sigmoidal, etc.
- *Sharpness* – a measure of the discontinuity in edge orientation at the point of the corner, being a continuum between a sharp change to rounded or blunted corners.

- *Colour* – a corner can be part of a light object on a dark background or *vice versa*. Since the foreground/background distinction problem is difficult colour can be measured instead relative to the region forming the smaller angle.

- *Junction Type* – determined by the number of regions converging at the corner, producing e.g. V, Y, ARROW, T, K, X, etc.

- *Size* – the range of scales over which a corner exists (i.e. Koenderink's [9] inner and outer scales).

Augmenting corners with this additional information makes them a much more powerful feature. When matching corner pairs the number of candidate corner matches can be considerably reduced and many ambiguities eliminated by requiring matching pairs to have similar attributes. Second, incorporating the corner's attributes can make it more selective for model indexing.

One approach to extracting the additional corner attributes would be to attempt to design a new corner detector with these capabilities. A relatively small number of mostly recent corner detectors do measure some of the above attributes. Li and Madhaven [10] use a set of basis functions for corner detection which also provides the orientation. Mehrotra *et al.* [12] detect half-edges from which the orientation and subtended angle can be calculated. Liu and Tsai [11] determine orientation and subtended angle, while Rohr [15] additionally provides blur (cf. edge profile) and contrast. However, both of these last two techniques are computationally expensive, involving model fitting requiring iterative solutions to non-linear systems of equations. Guiducci [7] also estimates orientation, subtended angle, blurring and contrast based on differential geometry. This requires the calculation of second order partial derivatives which are sensitive to noise. Few results are provided, but error high rates were reported (20% for contrast and 10% for the remaining parameters). Wang *et al.* used Beaudet's [2] DET operators to estimate corner sharpness, but did not analyse its accuracy.

Instead of incorporating property estimation within the detection stage, we take the approach of designing and applying a series of modules for extracting the attributes of corners that have already been detected. This has the advantage that existing corner detectors whose various merits (e.g. accuracy and robustness) are well known can be used [16]. Moreover, the modular approach to estimating the location and individual attributes of corners also enables a series of computationally efficient and simple solutions. In comparison, those methods that attempt to simultaneously estimate many parameters require more cumbersome and potentially less stable techniques. This paper describes several new methods for calculating the colour, orientation, subtended angle, and contrast of edges. The methods are assessed and compared by testing them on a set of synthetic corner images containing a variety of corner examples with varying degrees of added Gaussian noise and blurring.

2 Basis of Two Approaches

Two approaches are taken for calculating corner attributes. These are based on 1) intensity and gradient orientation histograms, and 2) local thresholding.

2.1 Intensity and Gradient Orientation Histograms

The histogramming approach uses a histogram of either the intensities or the gradient orientations of the pixels in the window centred on the corner. We use the Sobel operator to calculate the orientation and magnitude of the pixel gradients. Rather than apply a threshold to eliminate weak edges the orientation of each pixel is weighted by the magnitude of the gradient. For V shaped corners the orientation histogram should show two main peaks. The technique is not restricted to straight edged corners since curved edges will merely widen the orientation peaks. However, due to image noise, clutter, and blur, the histogram will be noisy. This is tackled by a multi-scale analysis of the histogram which performs the appropriate amount of smoothing to best distinguish the two peaks by merging fragmented peaks if necessary. A scale-space procedure could have been used [5]. However, a simpler method was developed instead which does not require explicit tracking of the peak over scale followed by parsing into syntactic units (fingerprints). Instead, the histogram is smoothed over a range of scales, and the scale maximising a heuristic measure of "2-peakness" is chosen. This measurement is the size of the two largest peaks less the sum of the sizes of the remaining peaks, and the whole normalised by the ratio of the second largest peak to the largest peak. For $i = 1 \ldots N$ peaks of size P_i and ordered in increasing size, this is given by:

$$M = \left(P_N + P_{N-1} - \sum_{i=1}^{N-2} P_i \right) \frac{P_{N-1}}{P_N}.$$

The term in brackets is intended to reduce the size of the subsiduary peaks while the factor that follows is intended to prevent the second largest peak being oversmoothed. Smoothing is performed by repeatedly convolving the histogram with the mask [0.2236, 0.5477, 0.2236], which provides a good approximation to a Gaussian function [4], until only two peaks remain. Further processing of the histogram is restricted to the scale that maximises M; the objective is to perform enough smoothing so that the main two peaks can be reliably identified while minimising the amount smoothing so as to reduce the displacement of the peaks. The only parameter required is the number of histogram bins (B) which was set to $B = 36$. Experiments showed that the algorithm is not sensitive to the value of B. For instance, both $B = 18$ and $B = 72$ produce similar results as shown in figure 1. Although the definition of M is heuristic a more formally based definition is difficult since this would require strict models for both the corner intensities and spatial distribution as well as the noise.

2.2 Local Thresholding

The second approach locally thresholds the image in a circular window centred on the corner to distinguish the foreground object that the corner lies on from the background. There are many thresholding techniques; we use Tsai's moment preserving method [19] which has been shown to perform relatively well [18].

3 Determining Attributes

3.1 Colour

To determine the colour of a corner we assume that there is a predominantly bimodal distribution of intensities in a window around the corner. The colour could be estimated by finding out which of the two main peaks in the smoothed intensity histogram is the larger. However, a more efficient method is to compare the intensity of the corner pixel with the other pixels in the window. If the corner pixel is lighter than the majority of the pixels then the corner can be classified as light, and *vice versa*. However, due to mislocalisation and blurring the corner pixel's intensity is unreliable. Instead, the following procedure uses all the pixel values within the window. The median value I_m and the average value I_a of the pixel intensities in the window are calculated. For a light corner the majority of pixels will be dark resulting in a dark median value. The average intensity combines both light and dark values, and will therefore be lighter than the median. The argument applies in reverse to dark corners.[1] Thus the classification is:

$$\text{corner} = \begin{cases} \text{light} & \text{if } I_m > I_a, \\ \text{dark} & \text{if } I_m < I_a. \end{cases}$$

Poor corner localisation can result in a change in the proportion of foreground and background pixels. This is reduced by increasing the window size; in the following examples a 7×7 window was used.

Alternatively, using the thresholding method the colour is given by the larger class produced by thresholding. Using Tsai's thresholding scheme it is unnecessary to explicitly count the size of the class since the appropriate p_0-tile for thresholding is already determined, and so:

$$\text{corner} = \begin{cases} \text{light} & \text{if } p_0 > 0.5, \\ \text{dark} & \text{if } p_0 < 0.5. \end{cases}$$

[1] The technique of testing the median intensity is related to the corner detector proposed by Paler *et al.* [13], who did not however use it to determine corner colour.

3.2 Orientation

The orientation of a corner is obtained by first performing the multi-scale analysis of the orientation histogram to find the orientation of the two main peaks. The corner's orientation is taken as the average of the two peak orientations. As an alternative to the multi-scale approach two single scale techniques, that are simpler and more efficient but less robust, are proposed. The first averages the orientations of all the edges in the window weighted by their magnitudes. The second returns the orientation of the single corner pixel.

In the thresholding method the orientation is determined as the angle of the vector from the centroid of the smaller class. This global approach (wrt the window) has the advantage over Brunnström *et al.* [3] that the boundaries of the corner do not have to be explicitly considered which is problematic if they are fragmented or curved.

3.3 Subtended Angle

Having calculated the multi-scale histogram the subtended angle is simply calculated as the difference between the orientation values of the two largest peaks in the histogram.

The subtended angle in the thresholding scheme is based on the size of the smaller class relative to the window area. Again this can be calculated directly from the p_0-tile value as:

$$\text{orientation} = \begin{cases} 360(1-p_0) & \text{if } p_0 > 0.5, \\ 360(p_0) & \text{if } p_0 < 0.5. \end{cases}$$

3.4 Contrast

The contrast of a corner is determined in a similar manner to the subtended angle except that the intensity histogram rather than the orientation histogram is analysed over multiple scales to find the two main peaks. The contrast is taken as the difference between the intensity values of the two main peaks.

Using the thresholding method the average image intensities in each of the two classes are calculated and contrast is taken as the difference of these two averages. Alternatively, the class medians can be used instead of averages in case noise and blurring across the edges biases the averages as estimates of typical intensities within the foreground and background regions.

4 Computational Complexity

Most of the methods for determining corner attributes are applied within a $w \times w$ window centred at the already detected corner where typically $w = [5, 11]$. Let $W = w^2$ equal the number of pixels in the window, then constructing the orientation or intensity histogram requires $O(W)$ time. The number of iterations required for smoothing the histogram can be assumed to be roughly independent of the window size. This was verified empirically by testing about 20000 corners.

The first stage of Tsai's thresholding algorithm, which is to determine p_0, is $O(W)$. The second stage to determine the appropriate threshold from the intensity range G approximating p_0 can be found using a binary search, and is $O(\log_2 G)$. Applying the threshold is also $O(W)$. Assuming that $O(W) \gg O(\log_2 G)$ ($G = 256$ in our examples) then both the histogramming and thresholding are $O(W)$. If a linear (i.e. $O(W)$) median finding method is used then, apart from the single pixel based orientation method which is $O(1)$, all the other methods for calculating colour, orientation, subtended angle, and contrast are $O(W)$.

5 Experimental Assessment

We described above a variety of methods for estimating the properties of corners. Here we show the results of testing these methods in a series of synthetic images, each containing one corner. Three sets of corners with subtended angles of 60°, 90°, and 120° were used, each at varying levels of contrast (50, 100, and 150 grey levels), added Gaussian noise ($\sigma = 50$, 100, 150, 200, and 250), and with and without Gaussian smoothing ($\sigma = 1$), making a total of 387 test corners. The results of the estimators of the corner properties were assessed at two locations, and their average performances are given for each. The first location is the ideal corner position of the uncorrupted test image (figure 2). The second was the closest pixel labelled as a corner by the Kitchen/Rosenfeld operator [8] (figure 3); this should provide a more realistic test since it incorporates typical corner mislocalisation. Comparing the accuracy of each method leads to the following conclusions for each property:

- *Colour.* With the well localised corners in the first test location both methods gave almost perfect results. However, given the mislocated corners in the second test, the median based method consistently outperforms the thresholding method.
- *Orientation.* The histogram and average edge orientation techniques performed better than the single corner pixel orientation method. The thresholding method worked well in the first test, but poorly in the second test, showing its sensitivity to corner mislocalisation.
- *Subtended Angle.* The thresholding method is again shown to be very sensitive to corner mislocalisation, and is therefore outperformed by the histogram based method.
- *Contrast.* The thresholding methods are consistently more accurate than the histogram approach, although the median histogram actually performs worse than just using the average class intensities.

In summary, the histogramming technique performs well for estimating the orientation and subtended angle, while the thresholding technique performs well for colour and contrast estimation. An alternative with similar accuracy to histogramming for calculating the subtended angle is the average orientation method.

The best of the above methods are demonstrated on the corners found by the Kitchen/Rosenfeld detector applied to the image of an indoor scene. The corner colours are shown in figure 4a. The majority of the classifications are correct but occasional errors are caused by very poorly localised corners. Although further increasing the window size may produce their correct classification by reducing the effect of mislocalisation there is the danger that other adjacent features in the image will be included in the window, confusing the classification. The subtended angles are shown in figure 4b. It can be seen that most of the marked corners corresponding to real corners in the scene have been assigned the correct properties.

6 An Example of Corner Matching

The usefulness of the additional attributes of corners is demonstrated by an application in which a widget is recognised and located in a cluttered scene. Matching is performed by detecting maximal cliques in an association graph (AG) [1]. The nodes in the AG are formed by all the pairwise combination of image and model corners that have the same properties. Pairs of nodes are linked as compatible if the corresponding pairs of image and model corners have the same binary properties. If corners are described by their position alone they have no unary properties that can constrain the number of possible model to image corner matches. The only possible binary property between two corners is distance (assuming the model orientation is unknown but the scale is fixed). Therefore, the AG will be large and densely connected, requiring substantial computation to find the maximal cliques, and probably also resulting in many incorrect matches. On the other hand, corners with attributes enable the subtended angle to be used as a unary constraint, and both distance and relative colour (assuming either that most of the object is lighter or darker than the background) are used as binary constraints.

The model widget is shown in figure 5; all angles are 90°. Since the corner detectors we use locate corners with angles less than 180°, the corners that form angles greater than 180° when tracing around the boundary are located on the background rather than the foreground. The boundaries of the widget (i.e. the straight lines) are not used in the matching process. The scene containing the widget and miscellaneous clutter is shown in figure 6a with the detected corners superimposed. The threshold on the difference in the values of the subtended angles was set to 20°, while no threshold was required for relative colour. The AG formed contains 107 nodes and 390 arcs, and the largest maximal clique contains five nodes which form the correct match. This match was used to determine the model transformation with the least squares error, which is shown superimposed on the image in figure 6b. The five corners that were matched are circled; the remaining corners were either not detected or had incorrect properties.

In contrast, for simple corners without the additional attributes the AG contains 864 nodes and 35696 arcs. Many spurious matches are generated with larger maximal cliques than the correct match.

7 Summary

This paper has asserted that many grey-level corner detectors are inadequate since they fail to extract many attributes which would improve the process of matching corners in two ways. First, corners could be eliminated from consideration if they did not satisfy certain criteria, thereby improving efficiency. Second, additional features could resolve otherwise ambiguous correspondences.

Several simple, non-iterative, and non-parametric techniques for determining several of the attributes of corners (i.e. colour, orientation, subtended angle, and contrast) were presented and tested. It was found experimentally that the histogramming method provides the best results for estimating orientation and subtended angle, while the thresholding technique performs best for colour and contrast estimation.

References

[1] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, 1982.

[2] P.R. Beaudet. Rotationally invariant image operators. In *Int. Conf. Pattern Recognition*, pages 579–583, 1978.

[3] K. Brunsström, J.O. Eklundh, and T. Lindeberg. On scale and resolution in active analysis of local image structure. *Image and Vision Computing*, 8:289–296, 1990.

[4] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31:532–540, 1983.

[5] M.J. Carlotto. Histogram analysis using a scale-space approach. *IEEE Trans. PAMI*, 9:121–129, 1987.

[6] R. Deriche and G. Giraudon. A computation approach for corner and vertex detection. *Int. J. Computer Vision*, 10:101–124, 1993.

[7] A. Guiducci. Corner characterization by differential geometry techniques. *Pattern Recognition Letters*, 8:311–318, 1988.

[8] L. Kitchen and A. Rosenfeld. Grey-level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[9] J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[10] X. Li and R. Madhavan. Correlation measures for corner detection. In *Conf. Computer Vision Pattern Recognition*, pages 643–646, 1986.

[11] S.T. Liu and W.H. Tsai. Moment preserving corner detection. *Pattern Recognition*, 23:441–460, 1990.

[12] R. Mehrotra, S. Nichani, and N. Ranganathan. Corner detection. *Pattern Recognition*, 23:1223–1233, 1990.

[13] K. Paler, J. Foglein amd J. Illingworth, and J. Kittler. Locally ordered grey levels as an aid to corner detetion. *Pattern Recognition*, 17:535–543, 1984.

[14] K. Rangarajan, M. Shah, and D. Van Brackle. Optimal corner detection. *Computer Vision, Graphics and Image Processing*, 48:230–245, 1989.

[15] K. Rohr. Recognizing corners by fitting parametric models. *Int. J. Computer Vision*, 9:213–230, 1992.

[16] K. Rohr. Localization properties of direct corner detectors. *Journal of Mathematical Imaging and Vision*, 4:139–150, 1994.

[17] P.L. Rosin. Acquiring information from cues. *Pattern Recognition Letters*, 14:599–609, 1993.

[18] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[19] W.H. Tsai. Moment-preserving thresholding: a new approach. *Computer Vision, Graphics and Image Processing*, 29:377–393, 1985.

[20] H. Wang and M. Brady. A practical solution to corner detection. In *Int. Conf. Image Processing*, volume I, pages 919–923, 1994.

[21] O.A. Zuniga and R. Haralick. Corner detection using the facet model. In *Conf. Computer Vision Pattern Recognition*, pages 30–37, 1983.

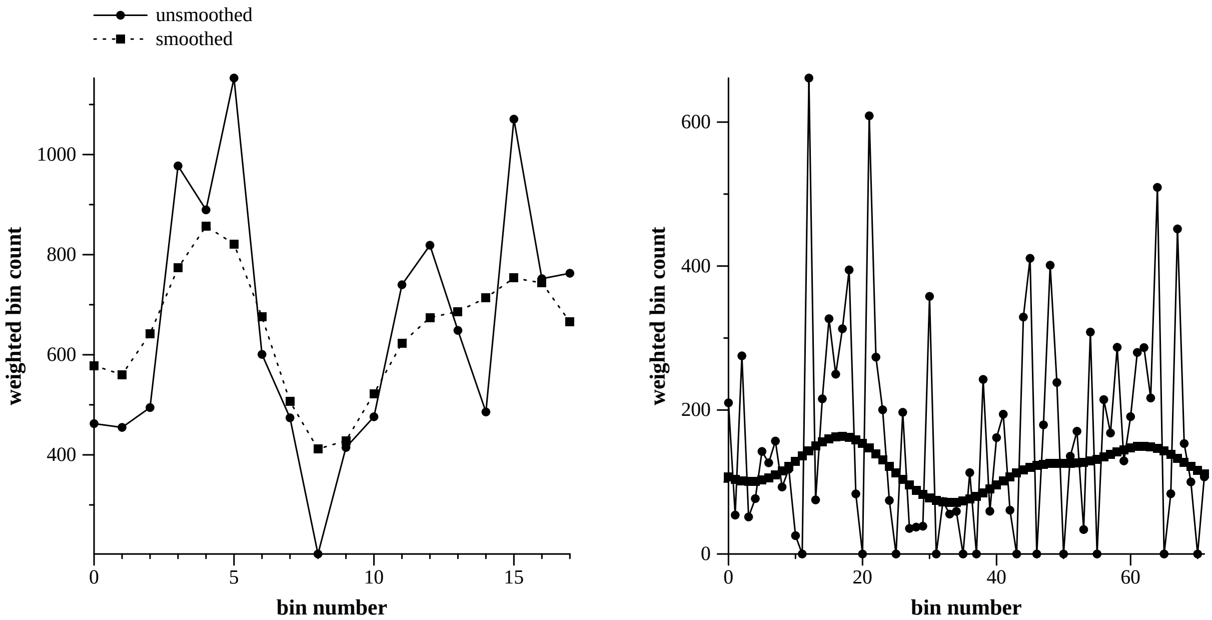
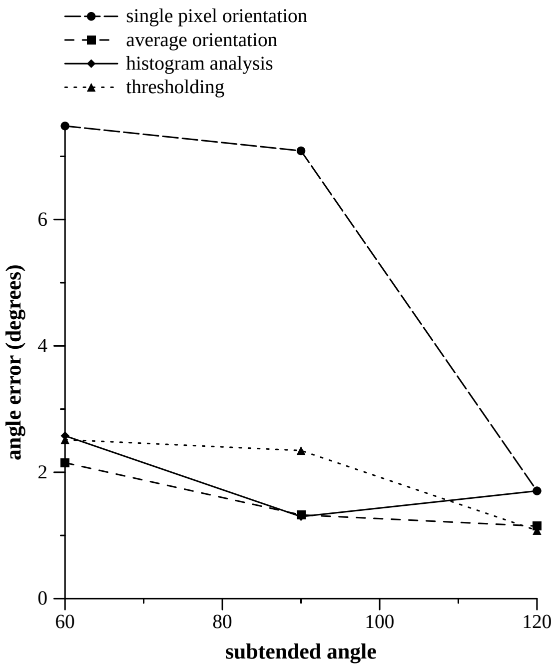
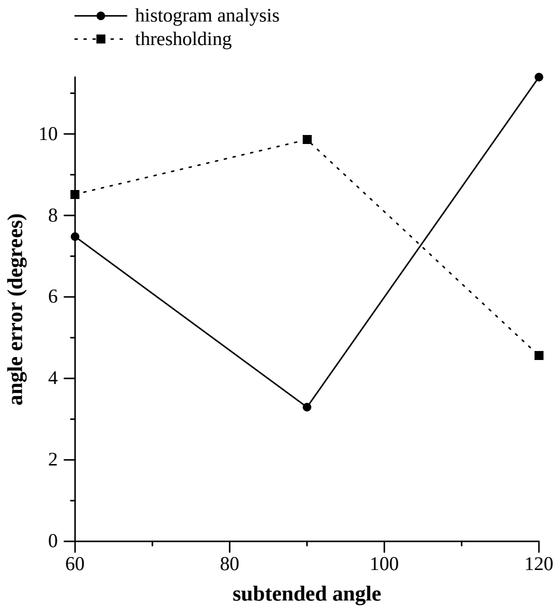


Figure 1: Automatic smoothing of histograms to identify two major peaks; changing the number of bins has little effect



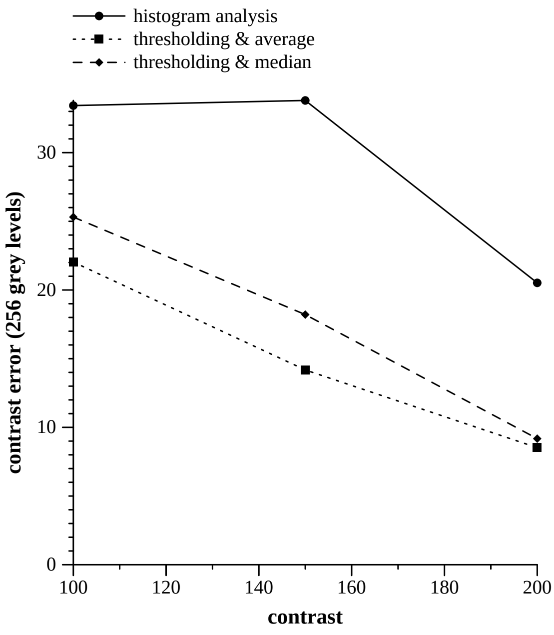
orientation accuracy

(a)



subtended angle accuracy

(b)



contrast accuracy

(c)

Figure 2: Performance accuracies of property estimation methods operating at the true corner position

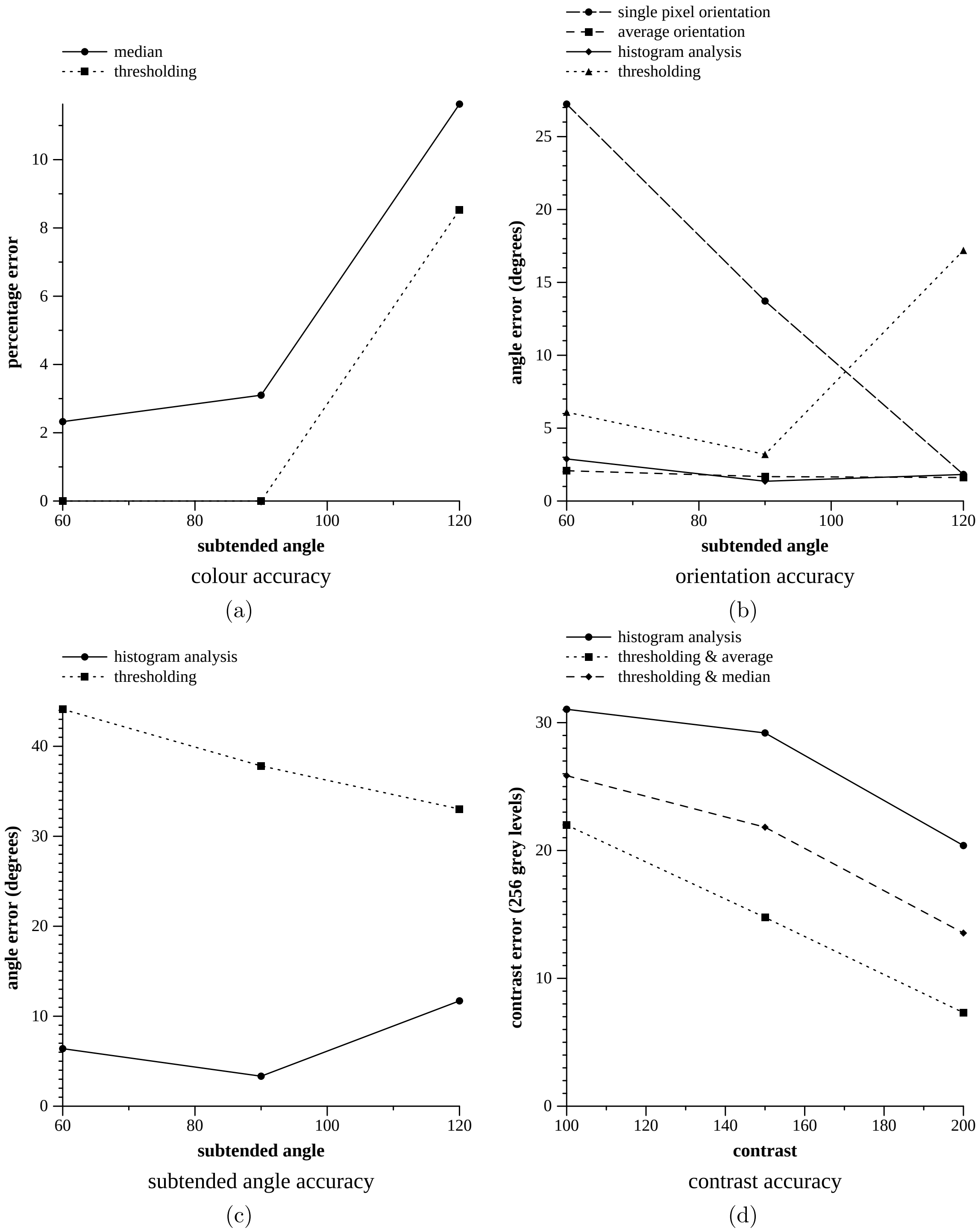
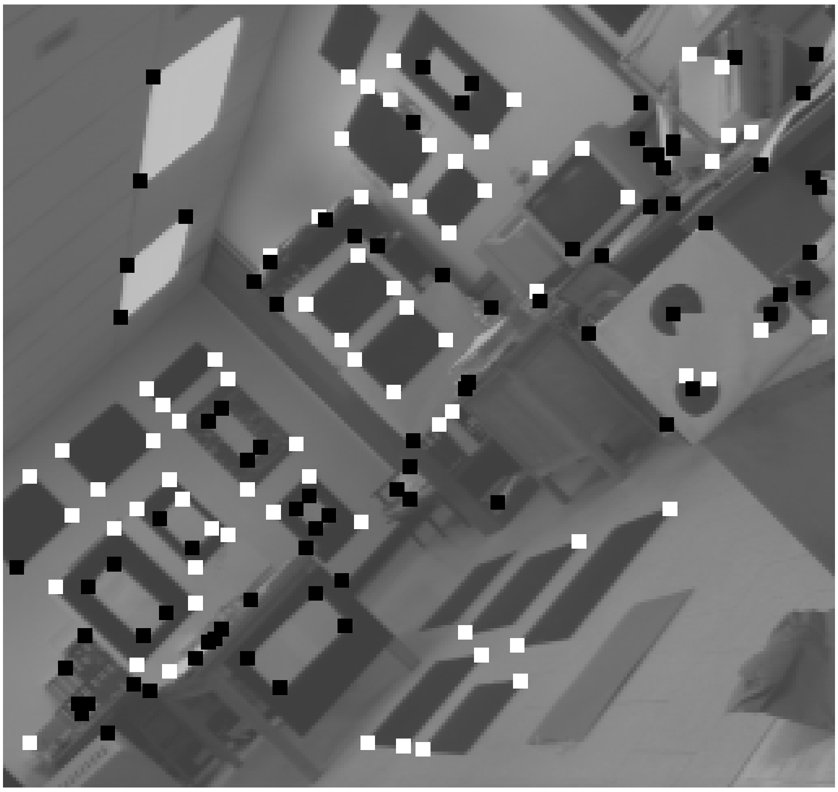


Figure 3: Performance accuracies of property estimation methods operating at the closest detected corner position



(a) Detected corners classified by colour – white squares mark dark corners and black squares mark light corners.

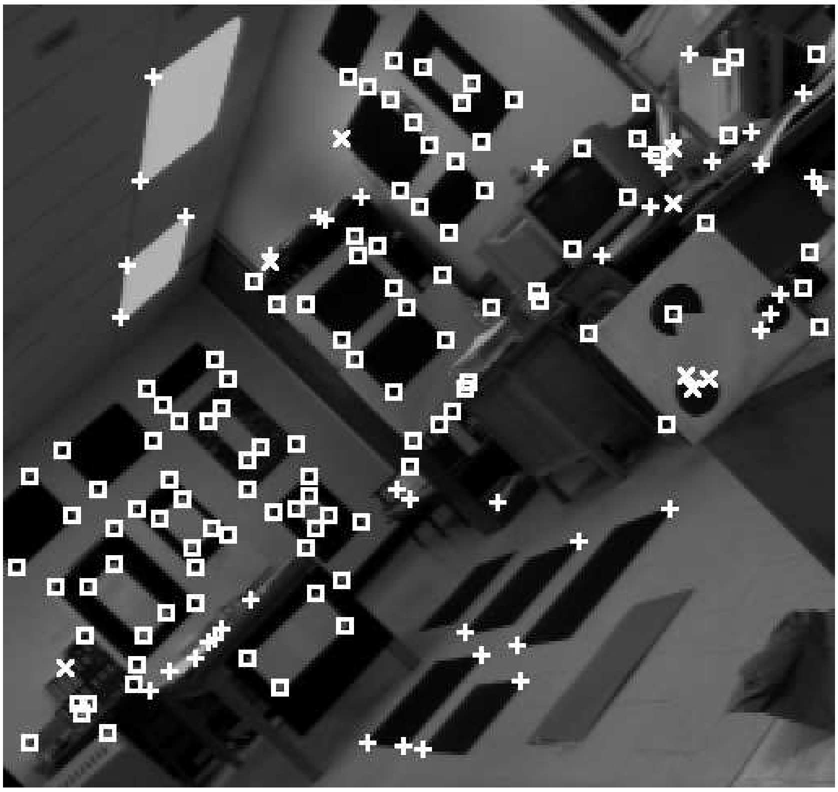


Figure 4: (b) Detected corners with subtended angles of $\pm 45^\circ \pm 22.5^\circ$ marked by horizontal crosses, angles of $\pm 90^\circ \pm 22.5^\circ$ marked by squares, and angles of $\pm 135^\circ \pm 22.5^\circ$ marked by diagonal crosses.

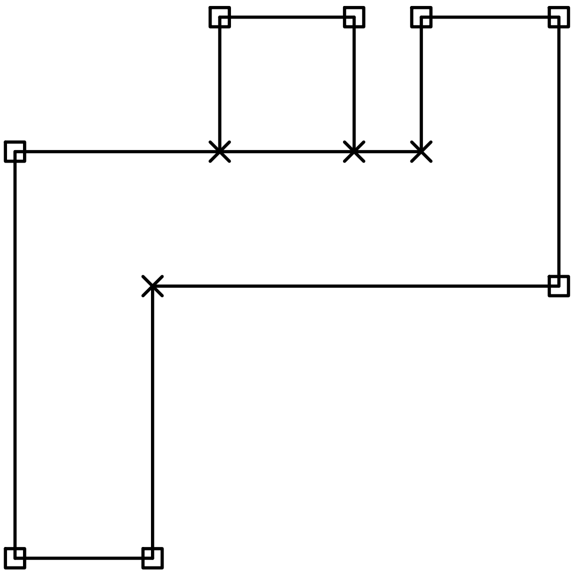


Figure 5: Model of widget identifying relative colour of corners (marked by boxes versus crosses)

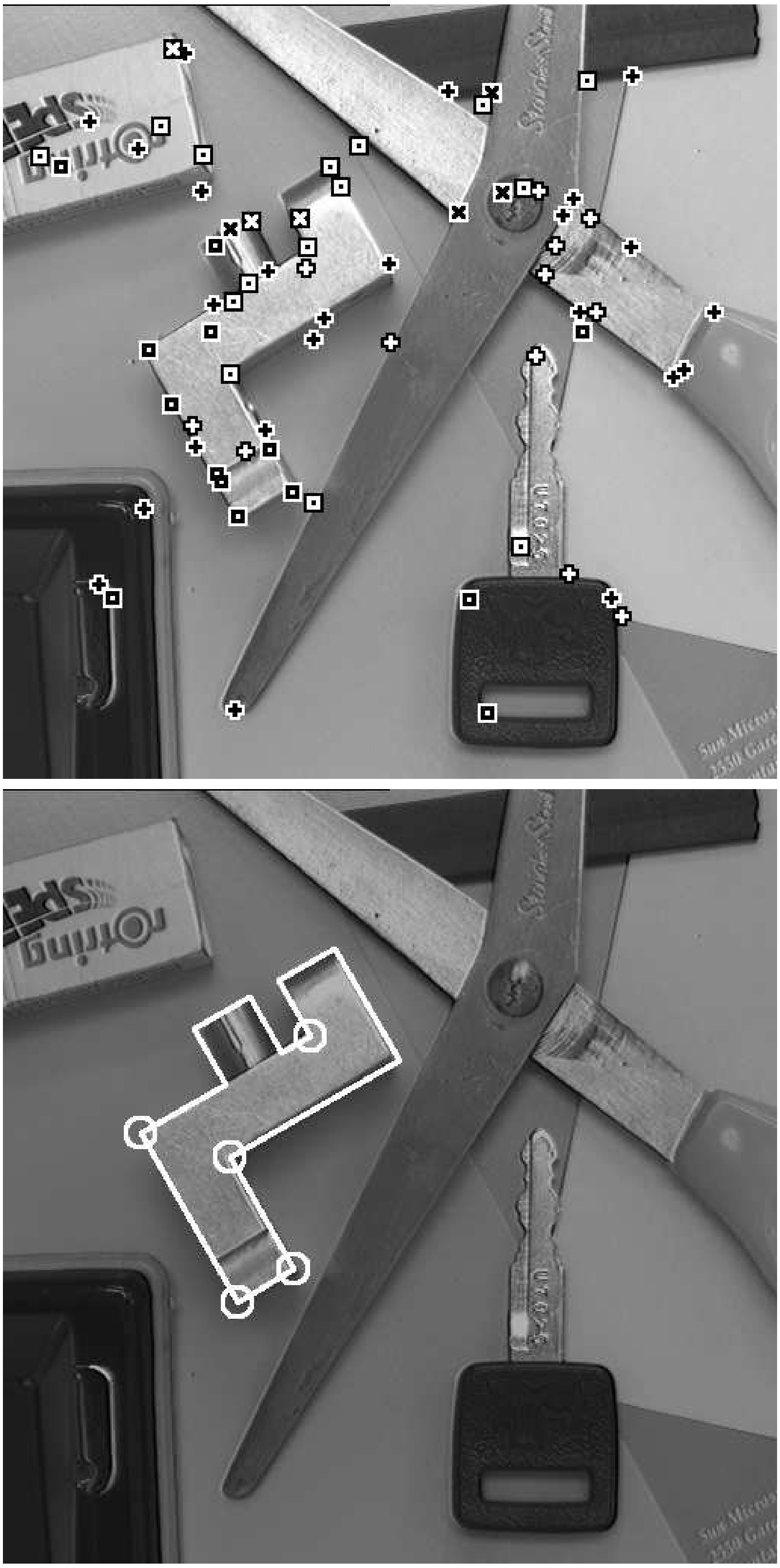


Figure 6: (a) Image overlaid with detected corners: dark/light corners marked white/black; subtended angles of $\pm 45° \pm 22.5°$ marked by horizontal crosses, angles of $\pm 90° \pm 22.5°$ marked by squares, and angles of $\pm 135° \pm 22.5°$ marked by diagonal crosses. (b) Matched and aligned model superimposed on image with matched corners circled