Shapes Fit For Purpose

Anupriya Balikai[1], Paul Rosin[2], Yi Zhe Song[1], Peter Hall[1]

Department of Computer Science[1], University of Bath, Bath BA2 7AY
School of Computer Science[2], Cardiff University, Cardiff CF24 3AA
(a.balikai | yzs20 | pmh@cs.bath.ac.uk)[1] paul.rosin@cs.cardiff.ac.uk[2]

Abstract

This paper is about shape fitting to regions that segment an image and some applications that rely on the abstraction that offers. The novelty lies in three areas: (1) we fit a shape drawn from a selection of shape families, not just one class of shape, using a supervised classifier; (2) We use results from the classifier to match photographs and artwork of particular objects using a few qualitative shapes, which overcomes the significant differences between photographs and paintings; (3) We further use the shape classifier to process photographs into abstract synthetic art which, so far as we know, is novel too. Thus we use our shape classier in both discriminative (matching) and generative (image synthesis) tasks. We conclude the level of abstraction offered by our shape classifier is novel and useful.

1 Introduction and Background

We wish to be able to describe any image using a collection of known shapes, specifically: ellipses, rectangles, triangles and convex hull when none of the others fit well. Fitting these shapes to image segments generates a description which offers a high level of abstraction which can be used in many applications. There is a precedent for choosing just these few simple shapes: both individual artists and Schools of Art in the early 20th century advocated the use of these shapes as basic constructs for painting. The artists found these shapes sufficient to model the visual world, producing both figurative and abstract painting. This provides *prima facie* evidence that these shapes make a powerful but simple descriptive set. We suggest they are useful too in Computer Vision, and provide two applications in support of this claim. One is processing photographs into abstract artwork, of possible value to the entertainment industry. The other is image matching. In particular we are able to match photographs to artwork, which could be of possible value in, say, content based retrieval applications.

In overview, our approach is as follows. First we segment an image, using N-cuts [4], chosen for its simplicity. The choice of segmentor is not important for shape fitting. Indeed, the shape fitter should operate independently of the segmentor. Second we optimally fit known shapes to each segment, and also robustly fit a convex hull [13]. The description is now ready for use in applications.

The shape fitting literature is large, so here we mention just a few examples. Shape fitting is usually restricted to a single shape model. For instance, circle fitting has been used for locating lunar craters [15] and soccer balls [18], ellipse fitting for face detection [20]

BMVC 2008 doi:10.5244/C.22.45

and the analysis of potatoes [23], superellipses for mines [5], rectangles for buildings [10], regular polygons for road signs [3]. But our task is to choose from amongst several shape families.

Fitting multiple shape models and then selecting the most appropriate is less common, and several problems arise. The first is that it is often convenient to fit different types of models using different error measures, which if they are not comparable cannot be used together as the basis of model selection. Second, the fitting errors from models of different complexity cannot be directly compared since the higher order models can always be expected to have a lower error of fit. Many schemes have been proposed to overcome this problem, one of the earliest being Akaike's information criterion (AIC) [1]. They operate by providing a measure that in addition to the fitting error combines and penalises the complexity of the model. For instance, AIC is defined as $-2\log(\text{likelihood}) + 2k$ where k is the number of parameters in the model. However, due to the different assumptions made by the various criteria regarding the distribution of the data, the different measures can give quite different results. For instance, Schwarz's Bayesian information criterion (BIC) [17], which is similar to the Minimum Description Length (MDL) criterion, penalises free parameters more strongly than AIC. Another criterion, the "geometric information criterion" was introduced by Torr [21], later the "geometric AIC" was suggested Kanatani [9]; both specifically designed for computer vision applications. Gheissari and Bab-Hadiashar [7] provide a review of such methods.

It is clear there is no single agreed way to fit some shape from more than one family. Issues of concern in the mathematical and computer vision literature are the robustness of the fit with respect to outlying data points, and the invariance of the fit under transformations of the data. The most common types of fitting in computer vision minimise some function (e.g. sum of squares) of the residuals. We note that measures are usually chosen for their mathematical tractability and computational convenience and complexity, rather than how well they correspond to perceptual or aesthetic judgements. Yet if we are to match photographs to human-made artwork, and to process a photograph into a synthetic artwork these value judgements are crucial. We have therefore opted to use a classifier which is trained under human supervision, in the hope to retain some degree of subjectivity..

Next is Section 2, which describes how we fit shapes from each of the families we have chosen, and also how to choose amongst these classes; some performance data for the classifier is given. In Section 3, we provide details of two applications: matching photographs to artwork, and automatically processing photographs into artwork. Finally we conclude the paper in Section 4 by observing that since our matcher operates well, and our synthetic art is of high aesthetic value, that our shape fitter is fit for purpose.

2 Method

We now describe how to fit a simple shape to an image region. Our account follows the order of our algorithm: first optimally fit several shapes, one from each of several classes; second choose amongst the optimally fitted shapes. It is the second step which is of the greater interest to this paper, since that is where novelty lies. The choice is made with a classifier, so this section concludes with some performance data to characterise the classifier.

2.1 Fitting Shapes of a Single Type

We fit four categories of shape. Three of them we call "known" because they can be qualitatively labelled: ellipse, triangle, and rectangle.

Voss and Süße described a powerful method for fitting a variety of geometric primitives by the method of moments [22]. The data is first normalised by applying an appropriate transformation to put it into a canonical frame. The fitted geometric primitive is then simply obtained by taking the geometric primitive in the canonical frame and applying the inverse transformation to it. For instance, for an ellipse they take the unit circle as the canonical form, and apply an affine transformation consisting of a translation to set the moments $m_{10} = m_{01} = 0$ and an anisotropic scaling such that $m_{20} = m_{02} = 1$. We have applied this approach to fit ellipses, rectangles and triangles.

The convex hull is an attractive symbolic representation of a shape on two counts. It is generally more compact (using only a subset of the original polygonal vertices), and also perceptually simpler since all indentations have been removed. However it has two limitations: it is insensitive to the size and shape of all indentations, and is also too sensitive to protrusions. To overcome these problems Rosin and Mumford [13] suggested a "robust" version of the convex hull, which is the convex polygon that maximises the area overlap with the input polygon. To compute the robust convex hull they used a genetic algorithm.

2.2 Selecting One Shape From Many

We are now able to optimally fit a collection of simple shapes to each region within a segmented image. The problem now is how to choose amongst them. Interaction is one approach, but not only is this tedious for the user but, we argue, is less interesting than considering automatic choice. Others have approached automatic selection through an information theoretic measure of some kind; Gheissari and Bab-Hadiashar [7] provide a review and an empirical comparison of these. As we have already observed, these measures are chosen for their mathematical tractability and computational convenience. But just as RMS between a decompressed image and its original is known to be a poor measure of subjective loss in quality, so these measures do not necessarily correspond well to human judgement of shape. It is reasonable to assert that using shape to match photographs to artwork, and indeed synthesising artwork from photographs, pre-suppose some level of human judgement. Therefore, we opted in favour of a trained classifier; training allows some subjectivity into the process. We now explain our classifier and the training regime.

Selecting appropriate shape models is done using a supervised classification paradigm. Specifically, a C4.5 decision tree [12] is learnt from a training set of regions which is then applied to new unseen data. The basis of a decision tree is that each feature can be used to make a decision that splits the data into smaller subsets, partitioning feature space into equivalence classes using axis-parallel hyperplanes. C4.5 builds decision trees by selecting the most informative feature (that is not yet considered in the path from the root) to split each subset. An entropy measure called normalised Information Gain [12] determines the effectiveness of each feature.

Regions are described by a feature vector and are manually labelled into shape categories. These features are the basis for making the decision regarding which is the most appropriate model. The feature vector consists of the errors between the region and each

of the fitted shape models. To compute the errors at each data point the shortest distance to the fitted shape is determined using the distance transform. However, the summed error is not a sufficient descriptor – it is easy to construct examples in which the best shape model (according to aesthetics and perceptual criteria) does not have a lower summed error. Instead the more information distribution of point errors is considered, and summarised by the following statistics: mean, standard deviation, skew, and kurtosis.

2.3 Performance Data

In this section we provide some performance data by which to judge the classifier. We restricted ourselves to training with the simple shapes, "Ellipse", "Rectangle", "Triangle", and (robust) "Convex hull". The training data came from N-cuts segmentations [4], we used 35 instances of each known shape to train. To test we used further data, again from N-cut segmentations, and so produced the confusion matrix in Table 1.

	ellipse	rectangle	triangle	convex hull
Ellipse	32	1	1	1
Rectangle	4	29	1	1
Triangle	3	1	30	1
Convex Hull	1	1	1	32

Table 1: The confusion Matrix, scaled by 35 (the number of instances per known class). Each row (capital letters) shows the result of classifying a ground truth set of a known shape. The fraction of times a ground truth shape class is classified as some nominated shape class is given. Each ground truth class contained N instances.

A natural question to ask is "how much confidence can one have in the confusion matrix?". Related to this is "when can training cease?". There is a common answer to these: cease training when the confidence matrix converges to a stable solution, so that one can have confidence it is "correct". Figure 1 shows how the maximum absolute change in any confusion matrix element (normalised by the total number of samples at each step) depends on the number of training data. It shows we can cease training after 35 or so training data per shape class.

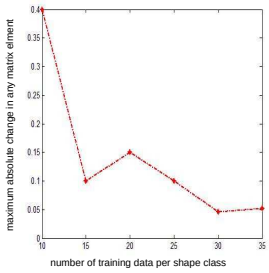

Figure 1: The maximum absolute change in any normalised confusion matrix element as a function of the number of training data in each shape class.

3 Applications

We have used our shape fitter in two applications: painting and matching. That these ostensibly distinct applications are related should not be too much of a surprise, since for a painting to be understood as representing some real world object one must be able to construct a mapping between them. Here we show we can synthesis art work from a photograph, and also match photographs to artwork. Taken together, these applications provide strong evidence of the utilitarian value of describing images using simple shapes.

3.1 Matching Photographs with Artwork

Our first application is matching photographs with artwork. This is of interest in further applications, such a content based retrieval. The problem with matching in this case is that the "character" of the two images can differ significantly. Paintings can comprise large regions of flat colour, photographs usually have far more detail in them than is necessary to convey the content, including complex light effects, textures and so on.

This problem has been sparsely addressed in the literature. Schechtman and Irani [16] assert that any textures can match, provided the are self-consistent. Bai *et al.* argue that structure is a class invariant [2]. Fidler and Leonardis learn tree structures premised upon Gabor filter responses [6] and use that tree to identify many specific objects classes; some supervised training is required at the higher levels of the tree. We continue the theme of abstract invariance by using qualitative shape as a matching primitive. This has the added advantage of allowing us to generate as well as classify, as the next application makes clear.

Our matcher accepts two images as input and returns a list of matched regions. The regions are produced by the N-cut segmentor [4]. This segmentor requires a single number, N, as input, and segments an image into that many segments. We use several values of N, specifically $N = 3i$, for $i \in [1,8]$ to obtain 8 "levels", each of finer granularity than the last. The nodes on the different levels generate a natural hierarchy — a tree — based on overlap.

Given two images we manually select a particular level from just one of them. This level is selected to give an acceptable segmentation of the foreground object of interest. This is reasonable, given the complete lack of prior information about what constitutes a semantic object. Our selections could be input to yet another classifier, which might then act in the manner of Fidler and Leonardis [6], but that is future work. This foreground object then acts as a query image. The matcher is to locate this object in the tree representing the second image.

The problem now is to find matches between two subtrees (actually, two forests), each of which corresponds to a foreground object in an image. Each node has a shape fitted to it, using the classifier, so also has a label which is an element of $\mathscr{L} = \{E,R,T,C\}$, corresponding to the four classes of training shape. The fact we use qualitative shapes means we wish to match through the shape labels that the classifier assigns. These we will call "observed" labels. The real, underlying shape label for a region is unknown to us — because the classifier may have assigned an incorrect label.

The rationale for using observed shape labels is that is provides a quick and easy way to match, and is naturally invariant to many geometric transforms, to clutter, and noise: matching photographs to artwork makes all of these demands. We used N-cut segments to

train, because we intended to use N-cuts to segment and match. Although we have opted to use qualitative data we nonetheless benefit from the a measure of the probability that two symbollic shapes match. We will estimate the probability that two observed. labels a and b, say correspond to the same underlying shape, the identity of which is never revealed. The confusion matrix in Table 1 plays a central role in this estimate.

Each row of the table gives the conditional probability $p(a|Z)$, which is the probability that a known *named* shape Z which is input to the classifier is assigned the *observed* label a. For example, $p(e|T)$ is the probability that a triangle is classified as an ellipse, e. We will continue to use upper case letters for known inputs to the classifier, and lower case for the labels it produces. Each row of the confusion matrix has constant names shape Z, each column has constant observed class a. Using Bayes' law we get the probability that a given observation a is really a named shape Z.

$$p(Z|a) = \frac{p(a|Z)p(Z)}{p(a)} \tag{1}$$

We know for sure that all named shapes exist atleast once we assume $p(Z) = c$ for all $Z \in \mathscr{Z}$. The probability of observing the label a requires us to marginalise over the named shapes:

$$p(a) = \sum_{Z \in \mathscr{Z}} p(a|Z) \tag{2}$$

Now suppose we have two shapes with observed classes (i.e. a name given by the classifier) a and b. The probability that these are both of the the same named shape Z is $p(Z|a,b)$. By appeal to conditional independence (and assuming statistical independence on the observations) we get

$$p(Z|a,b) = p(Z|a)p(Z|b) \tag{3}$$

The probability that the observed shapes a and b are the same underlying (but never revealed to us) shape is therefore

$$p(a,b) = \sum_{Z \in \mathscr{Z}} p(Z|a,b) \tag{4}$$

So $p(a,b)$ is a table entry that estimates the probability that two observed labels correspond to a named shape, matching in a qualitative sense.

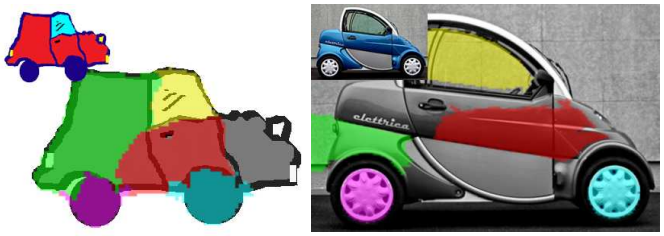

Figure 2: Parts of a drawn car and parts of a photographic of a car are matched, as shown by the colour coded regions.

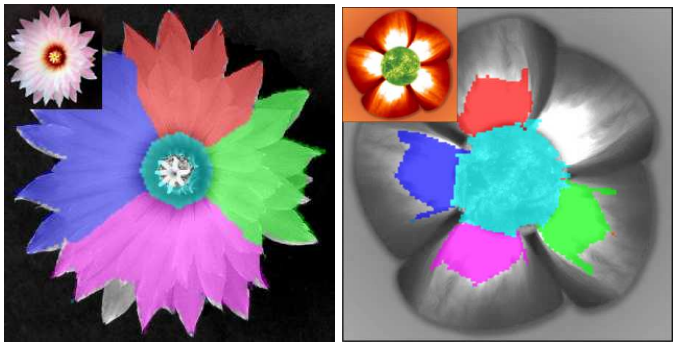

Figure 3: Parts of a flower are matched, as shown by colour coded regions.

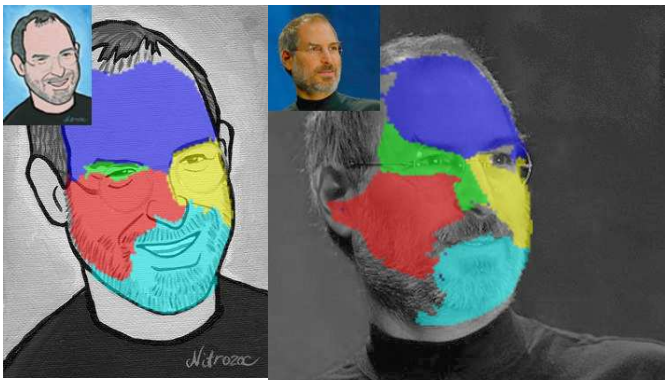

Figure 4: A painting and a portrait are matched, as shown by colour coded regions.

The probability table $p(a,b)$ is used to weight all matches between the regions in the two trees (forests). At the top-most level (the largest, coarsest regions) we consider all putative pairs of matches. We do likewise at the next level down, which expands each of the matched pairs. Recursive application generates a match-tree of all possible pair combinations. Match-tree branches are pruned where both children are not connected to both parents; ie. if (a,b) is a parent to (c,d) and (a,c) are connected in image one, then (b,d) must be connected in image two. We seek the path of maximal probability in this tree. A path comprises a string, $\mathscr{S}$ of matched pairs, each region appears at most once in such a string. The probability of the path is then $p(\mathscr{S}) = \Pi_{(a,b)\in\mathscr{S}} p(a,b)$. Since paths can be of different lengths we normalise to take the geometric average, so $p(\mathscr{S})^{(}1/|\mathscr{S}|)$. This is equivalent to a "characteristic" radius of a $|\mathscr{S}|$ dimensional hyper-ellipse whose main axis radii are the probabilities along the path.

Results from some of our matched photograph/painting pairs are shown in Figures 2, 3, and 4. The matcher has successfully matched corresponding regions in these images, even where colour and and other properties differ significantly. It has not succeeded in matching as well as would wish. At the moment our explanation lies with the N-cut segmentor, because it can be unreliable. Despite this, these results indicate that qualitative shape can be used to as the basis of a matcher. Beside matching across photographs and

artwork, a qualitative matcher such as ours might be used to initialise a more complex, quantitative matcher. Or, the same matcher as ours might be adapted to include measurement data: e.g. how similar is a region to each of the shape classes; or the convex hull could be used as a kind of "wild card" on the grounds that something classified as that could in principle be just about any shape.

The point of this matcher, though, was to explore the possibility of using nothing but qualitative data. Our results show that even such weak measures can prove useful.

3.2 Synthetic Abstract Art from Photographs

In our second application, we use the shapes fitted by the same classifier as used by the matcher to create synthetic artworks of an abstract nature. Specially, the types of abstract artworks we produce are largely motivated by artists such as Kandinsky and late Matisse, who used pure geometric shapes as primitives to create art. Two representative artworks are "Several Circles, 1926" by Kandinsky, where objects are represented as a collection of circles and "The Snail, 1952-53" by Matisse, where he used a collection of paper cut-outs to create a snail. For copyright reasons, examples of such artworks can not be shown here, but can be readily found on-line. Our classifier allows the synthetic art to be created which is more abstract in nature than is typical — most of the literature concentrates on making marks [8, 14] rather than producing abstraction, although the field is moving in that direction [11].

We start by segmenting the image into different granularities using the same segmentor used in the matcher, except that only two levels of granularity are sufficient in this application. Specifying these two levels is the only user interaction we require, and these two remain valid for many images, typically $N = 50$ for fine detail, and and $N = 5$ for coarse background, are input to N-cuts [4]. So, in many cases the user just needs to specify which image is to be processed.

The finer segments are rendered on top of the coarser ones, but only after filtering out some of the detailed shapes; otherwise too much detail is shown. To filter non-salient detail we use colour differencing. The colour of the fine segment is compared to that of the coarse segment it overlays. Colour differences are measured in terms of just noticeable difference (jnd) in CIELAB colour space. For instance, colours, (L_1, a_1, b_1) and (L_2, a_2, b_2), have colour difference ΔE_{12} as follows

$$\Delta E_{12} = \frac{\sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}}{jnd} \tag{5}$$

where $jnd \approx 2.3$ in CIELAB colour space [19]. By thresholding ΔE we can control the level of detail to render on the top layer; increasing the threshold results in less shapes being rendered and vice versa. A constant ΔE value of 5 is used to make all rendering in this paper.

Order matters rendering shapes into a frame-buffer, because shapes fitted to regions at a single N often overlap. We tackled this problem by introducing a shape fitting error τ. Given a shape model S and its corresponding region R. τ is defined as the following ratio, $|S \bigcap R)|/|S \bigcup R)|$, which is a form of Tanimoto similarity score, calculated on a per pixel basis. Shapes with large fitting errors are rendered before those with smaller errors. To create an embossed look, to the paper cuts we counted the number of shapes lying over

each pixel; the resulting height field became a bump map. To create transparent paper we simply used “alpha” colour channel. Figures 5 and 6 show some sample output.

Figure 5: Left, the photograph car in Figure 2 has been rendered as paper cut-outs, which show the shape fitted to each region. Right, a photograph of a flower is rendered, this time as transparent shapes.

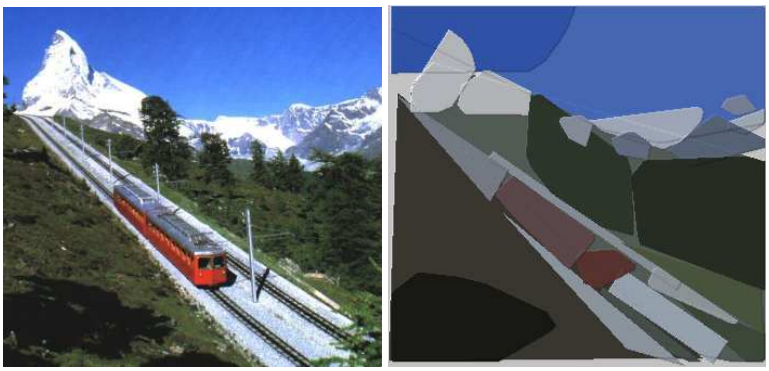

Figure 6: Left, an original photograph, right an abstraction in shapes. This shows that simple shapes are sufficient to capture the essence of a complicated image: the lack of detail can be advantageous and even desirable.

4 Discussion and Conclusion

This paper provides a novel method to fit not just a single shape to a region, but a way to classify a region as some shape from amongst several families. The classifier is extensible to shapes other than those we have chosen here — super-ellipses can be classified too, for example. We restricted ourselves to simple shapes based on the precedent of early 20th century art.

The descriptions in images that come from our classifier have been put to use in both discriminative tasks (matching) and generative tasks (synthesis). Both applications offer novelty and both could find use elsewhere, so are utilitarian too. We conclude that a shape based image description offers a level of abstraction that is of value to computer vision.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–785, 1974.

[2] X. Bai, Y.Z. Song and P.M. Hall. Learning object classes from Structure. In *BMVC* 172–181, 2007.

[3] N. Barnes, G. Loy, D. Shaw, and A. Robles Kelly. Regular polygon detection. In *ICCV*, pages I: 778–785, 2005.

[4] T. Cour, F. Benezit and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR* 1124–1131, 2005

[5] E. Dura, J.M. Bell, and D.M. Lane. Superellipse fitting for the classification of mine-like shapes in side-scan sonar images. In *IEEE Conf. Oceans*, volume 1, pages 1: 23–28, 2002.

[6] S. Fidler and A. Leonardis. Towards Scalable Representations of Object Categories: Learning a Hierarchy of Parts. In *ICCV*, 2007.

[7] N. Gheissair and A. Bab-Hadiashar. Model Selection Criteria in Computer Vision: Are They Different? In *Digital Image Computing: Techniques and Applications*, 185–194, 2003

[8] P. Haeberli. Paint by Numbers: Abstract Image Representations. In *SIGGRAPH* 207–214, 1990

[9] K. Kanatani Uncertainty Modeling and model Selection for Geometric Inference. *IEEE TPAMI* 26(10), 1307–1319, 2004.

[10] M. Ortner, X. Descombe, and J. B. Zerubia. A marked point process of rectangles and segments for automatic analysis of digital elevation models. *IEEE TPAMI*, 30(1):105–119, 2008.

[11] A. Orzan, A. Bousseau, P. Barla and J. Thollot. Structure Preserving Manipulation of Photographs. In *NPAR* 103–110, 2007.

[12] J.R. Quinlan C4.5: progams for Machine Learning. *Computer Vision and Image Understanding* 103(2), 101–111, 2006

[13] P.L. Rosin and C.L. Mumford. A symmetric convexity measure. *Computer Vision Image Understanding* 103(2), 101-111, 2006.

[14] M.P. Salisbury, S.E Anderson, R. Barzel and D.K. Salesin Interactive Pen-And-Ink Illustration In *SIGGRAPH* 101–108, 1994

[15] Y. Sawabe and T. Matsunaga S. Rokugawa. Automated detection and classification of lunar craters using multiple approaches. *Advances in Space Research*, 37(1):21–27, 2008.

[16] S. Schechtman and M. Irani. Matching Local Self-Similarities across Images and Videos. CVPR, 2007.

[17] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[18] C.J. Seysener, C.L. Murch, and R.H. Middleton. Extensions to object recognition in the four-legged league. In *RoboCup*, volume 3276 of *LNCS*, pages 274–285, 2004.

[19] G. Sharma Digital Color Imaging Handbook *CRC press*, 2003

[20] Q.B. Sun, C.P. Lam, and J.K. Wu. A practical automatic face recognition system. In H. Wechsler, J.P. Phillips, V. Bruce, F.F. Soulié, and T.S. Huang, editors, *Face Recognition, From Theory to Applications*, pages 537–546. Springer, 1998.

[21] An Assessment of Information Criteria for Motion Model Selection. in *CVPR* 47–53, 1998.

[22] K. Voss and H. Süsse. Invariant fitting of planar objects by primitives. *IEEE TPAMI*, 19(1):80–84, 1997.

[23] L.Y. Zhou, V. Chalana, and Y. Kim. PC-based machine vision system for real-time computer-aided potato inspection. *Int. J. Im. Systems and Tech.*, 9:423–433, 1998.