

# How does the Hue contribute to construct better colour features?

Giovani Gomez Estrada<sup>1</sup> and Eduardo Morales<sup>2</sup>

<sup>1</sup> Max-Planck-Institute, Heisenbergstraße 3, D-70569, Stuttgart, Germany, [giovani@mf.mpg.de](mailto:giovani@mf.mpg.de)

<sup>2</sup> ITESM Cuernavaca, Reforma 182-A, Temixco, 62589, Morelos, Mexico, [eduardo.morales@itesm.mx](mailto:eduardo.morales@itesm.mx)

**Abstract.** We explore the impact of including hue in a feature construction algorithm for colour target detection. Hue has a long standing record as a good attribute in colour segmentation, so it is expected to strengthen features generated by only RGB. Moreover, it may open the door to infer compact feature maps for skin detection. However, contrary to our expectations, those new features where hue participates tend to produce poor features in terms of recall or precision. This result shows that (i) better features can be constructed without the costly hue, and (ii) unfortunately a good feature map for skin detection is still elusive.

## 1 Introduction

It is well-known that pixel based skin detection plays an important step in several vision tasks, such as gesture recognition, hand tracking, video indexing, face detection, and computer graphics, see e.g. [3, 4, 9, 14, 23, 25]. Although colours may vary due to ambient light, brightness, shadows or daylight, skin detection is computationally tractable. Practitioners had traditionally used just existing off-the-shelf colour models, like HSV, YUV, raw RGB or normalised RGB, which however often yields poor precision<sup>3</sup> to upper layers in their vision systems.

We know there is no single colour model suitable for skin detection [6], and that traditional models are not so useful [1, 21] for this task. Fortunately, machine learning community had developed the concept of attribute construction, which in this context means to override all existing attributes and infer new ones for the task at hand. However, *which attributes should be used to infer a good model for skin detection?*

We present a computational study where R, G, B and hue participate within an attribute construction approach. Guided by existing literature, we expect a clear improvement over features generated with only R, G, and B. Hereafter, we shall refer to “attribute” as raw input variables, and “feature” to any combination of attributes.

---

<sup>3</sup> Precision =  $\frac{TP}{TP+FP} \times 100\%$ , where TP = true positives; and FP = false positives, where the prediction is incorrectly set as *skin*

This report will review, in section 2, the crossroads of colour detection and machine learning. Then, in section 3, the main premise and expectations are presented. Two influential tools from machine learning, attribute construction and attribute selection, are described in sections 4 and 5, respectively. Experimental settings and findings appear in section 6, while conclusions to this work are found in section 7.

## 2 Improving colour detection

Although eliminating the illumination components has been a popular practice for skin detection, it has not really improved our vision systems. What is even worse, it has been reported that this practice actually decreases the separability in some models [21]. The other way around, adding components in a stepwise procedure have the same problems. For instance, what you gain with  $C_r$  or hue is lost by adding its second element,  $C_b$  and Saturation, respectively. In this direction there are some bad news, see e.g. [1, 21], that many practitioners have sadly noticed, no matter which existing colour model we use, the separability of skin and non-skin points is independent of the colour model employed.

Of course, we may argue that a long term solution would be to look at different wavelengths. But, for the time being, red-green-blue response prevails as the standard input value. Thus, if existing general colour models do not help for target detection one may be tempted to create *synthetic* ones. Say, these may not be reversible, not useful for other tasks, nor adding anything new to Colour Science. Of course, those synthetic models have not to be created by guessing combinations in unsound way but with a systematic procedure. In general, any feature space tuned for this purpose will hereafter be cited as “skin colour model”.

### 2.1 Learning and colour

Since our raw attributes are not very helpful for skin and non-skin discrimination, we may therefore transform RGB into a nonlinear mapping. Projecting RGB into higher spaces may result in easier decision boundaries. A line in these new spaces is in fact a non-linear one in the original space. It is known from Statistical Learning Theory that a hypothetical polynomial of sufficiently high degree do approximate any arbitrary decision boundary, no matter of how complex it is. Nevertheless, for our purposes two practical questions arise, (1) how do you create such a non-linear features in a constructive manner, and (2) how do you get *only few* of them to describes a good decision boundary.

Two recent steps have been done to bring machine learning approaches to help colour detection [5, 29]. While in both cases colour targets were fed into a decision tree induction system (c4.5 [19]), the main difference is the attribute construction step. In [29] the RGB stimulus was directly transformed into hue, saturation, and average values of R, G, and B. Resulting decision trees use those attributes, e.g. moment of inertia, to classify colour targets.

Since their work uses pre-existing models, it is more in the sense of the Michalski's view of pattern discovery in computer vision [15]. However, under this idea, we assume that the current description language is good enough to describe our colour targets. A different path was adopted in [5], and no pre-existing models were assumed. Then, as their induction systems progress, new and more specific features are constructed. In each cycle, current features are passed to an attribute selection step, and only those which exhibit good performance are allowed to continue in the process. Although both approaches are bit costly in terms of computing time (e.g. attribute selection and tree induction), it is worthwhile to explore machine learning ideas to automate colour detection.

### 3 Why not hue?

As mentioned before, authors in [5] did use RGB to create new features. However, we may criticise why they did not include hue in their initial set. Hue has a long standing record of good colour attribute. Moreover, it has been recently assessed as one of the most influential attributes in a survey [6]. Thus, it seems quite normal to include hue and propose two questions:

1. does Hue contribute to infer better colour spaces for skin detection?
2. whether or not is possible to infer a 2D skin space.

The intuition behind adding hue is far clear. Hue may be used as a shortcut to get more compact features. It is an obvious candidate for any induction system. Many other colour attributes are easily derived as a linear combination of RGB, meanwhile Hue is quite more complex to infer. By allowing hue to participate in the attribute construction, one may expect powerful features with better recall, precision and success measures<sup>4</sup>. Further, as a by-product, one may therefore expect to find out a good 2D model. Many current approaches to skin detection use three components (e.g. [5, 6, 14]), or a 2D where the illumination one has been just removed.

### 4 Attribute construction

Most constructive induction systems use boolean combinations of existing attributes to create new ones, e.g. [18, 20, 27, 28]. Say, their constructive operators can form conjunctions and/or disjunctions of attributes (e.g. [11, 12, 18, 20]) or even use more sophisticated operators such as M-of-N [16] and X-of-N [27]. M-of-N answers whether at least M of the conditions in the set are true. X-of-N answers how many conditions in this set are true. Although a large number of studies have been devoted to boolean combinations of attributes (e.g. [28]), there are very few systems that use arithmetic combinations of real-value ones,

<sup>4</sup> Recall =  $\frac{TP}{TP+FN} \times 100\%$ , where TP = true positives.

Success rate =  $\frac{TP+TN}{TP+FP+TN+FN}$ , where FN = false negatives.

which normally occur in vision. Most notably is the Bacon system [13] which searches for empirical laws relating dependent and independent variables. Bacon finds increasing and decreasing monotonic relations between pairs of variables that take on numeric values and calculates the slope by relating both terms to create a new attribute. Once a functional relation between variables is found, it is taken as a new dependent variable. This process continues until a complex combination is found relating *all* the primitive attributes.

In this paper we start with hue and the three basic color components RGB in a normalised form, and a simple set of arithmetic operators to produce a suitable model for pixel based colour detection. Once a new set of attributes is produced, a restricted covering algorithm (RCA, [5]), is used to construct single rules of no more than a small number of easy to evaluate terms with a minimum accuracy. We are interested in inducing simple models as they are relevant to applications which require fast response times, such as, semi-automatic calibration of colour targets, gesture recognition, face and human tracking, etc.

The general approach followed in this paper for constructive induction is shown in Table 1. The idea, is to start with some primitive attributes and a set of constructive operators, create a new representation space, run an inductive learning algorithm, and select the best attributes of this new space. This process continues until a predefined stopping criterion.

CurrentAttrib = original attributes, i.e.  $\{\frac{r}{r+g+b}, \frac{g}{r+g+b}, \frac{b}{r+g+b}, Hue\}$

Operators = set of constructive operators, i.e.  $\{+, -, *, /\}$

UNTIL termination criterion

- NewAttrib = CurrentAttrib  $\cup$  new attributes  
constructed with Operators on CurrentAttrib
- Run a machine learning algorithm on NewAttrib
- CurrentAttrib = Select the best attributes  
from NewAttrib

**Table 1.** General constructive induction idea. (i) the machine learning algorithm, (ii) the constructive induction module, and (iii) an evaluation component.

The constructive induction algorithm starts with hue,  $\frac{r}{r+g+b}$ ,  $\frac{g}{r+g+b}$ , and  $\frac{b}{r+g+b}$ . All of them were used to create new attributes by seven constructive operators:  $A + B$ ,  $A * B$ ,  $A - B$ ,  $B - A$ ,  $A/B$ ,  $B/A$ , and  $A^2$ , where  $A$  and  $B$  can be any pair of distinct attributes.

## 5 Attribute selection

While it is relatively easy to create new features, their evaluation is a very time consuming step. It is the internal loop in these induction systems, say, every new hypothesis or features have to be assessed in their goodness to discriminate the

target classes. There are basically two main approaches in attribute selection: filters and wrappers. Filters rank variables independently of their later usage. Conversely, wrappers guide their attribute selection process specifically for the machine learning algorithm employed. Generally speaking, filters are faster than wrappers. However, it is commonly assumed that wrappers do offer better predicting performance, i.e. the selected subset is tightly tuned for the machine learning algorithm and thus for the predictor too. Additional information on attribute selection can be found in one recent survey [8], and a special issue on Variable and Feature Selection in [10].

In this paper we adopt a wrapper approach using an information gain heuristics. The resulting representation is a tree-like structure, generated by RCA, which is chosen because of two main advantages: (i) simplicity in both representation and computing requirements, and (ii) able to produce a range of models for *the target* class. The algorithm is briefly described within the next paragraphs.

### 5.1 RCA

The general strategy of RCA is to favour attributes which cover a large number of true positives and attributes with small number of false positives. We are interested in single rules, so we shall talk about the total number of true positives (*TTP*) which will be used to increase the measure of recall and the total number of false positives (*TFP*) which will be used to increase precision.

Since we are dealing with real-value attributes, RCA creates binary splits using an information gain heuristics (as C4.5 does, [19]). RCA considers two possible attributes in parallel when constructing rules. On its first cycle, RCA constructs two rules which have as LHS the attribute with larger *TTP* in one rule and the attribute with larger  $TTP - TFP^2$  in the other rule. The following cycle produces two rules out of each original rule (4 in total) following the same criterion, again adding to the LHS of each rule one attribute with large coverage and one which is heavily penalized by the number of misclassifications. This process continues until the rules produced have a certain number of predetermined terms. The upper bound of rules to be produced is  $2^n$ , where  $n$  is the number of terms on each rule. RCA builds  $2^n$  rules in parallel aiming for a large coverage with small errors on the same example set. This idea handles two objectives, thus it produces a range of alternatives, potentially from *high recall-poor precision* to *high precision-poor recall*, and balanced intermediate states, of course. An overall description of RCA is given in table 2.

### 5.2 Connection to other methods

The idea of improving features on-the-fly is not new. To our best knowledge, it appeared in the MOLFEEA project, an inductive database system, e.g. [11, 12]. However, they use a boolean conjunction of predefined attributes to generate new ones. We share the aim to generate features not in advance but “on demand”, by detecting the need for a representation change, but nevertheless we do use arithmetic instead of boolean combinations.

**For each** class  $C$   
 Let  $E$  = training examples  
 Let  $N$  = maximum number of terms per rule  
 Create a rule  $R(0)$  with empty LHS and class  $C$   
 Let depth  $D = 1$   
**Until**  $D = N$  **do**  
   **For each** attribute  $A$  create a split ( $Sp_A$ )  
     with greater information gain  
   **For each** existing rule  $R(D - 1)$   
     create two new rules ( $R_1(D)$  and  $R_2(D)$ ) by  
     adding to its LHS, a  $Sp_i$  with larger  $TTP$   
     ( $R_1(D)$ ) and a  $Sp_j$  with larger  $TTP - TFP^2$   
     ( $R_2$ )  
   Let  $D \rightarrow D + 1$   
   **For each**  $R_i(D)$  continue with its own  
     covered examples from  $E$   
 Output all  $R_i(D)$

**Table 2.** Overall description of RCA. Two intermixed criteria to induce rules with complementary attributes.

Other works [17, 22] did use genetic programming as a preprocessing step. They construct new features from the initial dataset, generating “potentially” useful features in advance. A genetic algorithm is then used to control the attribute selection step. Here, a binary chromosome represents attributes, say, the 0/1 or (false/true) slot means whether the corresponding attribute does participate in the wrapper process. This idea does not introduce any bias in the feature construction in a class-dependent or goal-driven fashion, as previously mentioned.

Although an open avenue would be to use other (e.g. evolutive) ideas for attribute construction and selection, which may suits well in this context, we feel that our proposed technique is rather straightforward (see table 2) and avoid unnecessary costly operations.

## 6 Experimental settings and results

We create ten subsets with skin and non-skin elements. The dataset is described in [6], which is based on real skin images, from different input sources, illumination conditions and races, with no photographic manipulation. For each data set, we selected 33000 skin and 67000 non-skin elements uniformly at random. We perform 10-fold cross validation in the attribute selection step, the inner loop. The covering algorithm, RCA, wraps c4.5 for attribute selection, which ran with the usual pruning and confidence thresholds. In addition, we request a minimum number (500) of elements per leaf to force fewer leaves. Final results of each induced model were calculated on large and balanced unseen data: 12.2 million points for each skin and non-skin targets.

Table 3 shows the best bi-dimensional models found by RCA. RCA is designed in such a way that one feature can appear twice in the same branch. Hence, it introduces a double threshold. Only two leaves were generated with this concept. To our surprise, the best precision on both 2D and 3D models is the first model in table 3. Normally, what is expected is a high recall, but in this case  $\frac{gb}{(r+g+b)^2}$  and hue contributed to achieve 92.6% of precision.

colour models (two components)		recall (%)	precision (%)	success rate (%)
$\frac{gb}{(r+g+b)^2}$	h	80	<b>92.6</b>	86.3
$\frac{gb}{(r+g+b)^2}$	$\frac{h*(r+g+b)}{b}$	97	74.8	82.7

**Table 3.** Only two models had two variables, yet one of them has the higher precision among all experiments.

The same features in table 3 were selected in other branches, as shown in first two lines of table 4. Interestingly, better models were found by removing these double thresholds and selecting a new variable instead. Thus, the overall effect does eliminate a threshold on  $\frac{gb}{(r+g+b)^2}$  and increases the recall. We should be critic with the second row in table 4, in which recall is rather high. It is not a surprise since even a plain ratio like  $\frac{r}{g}$  can achieve more than 95% in recall, but at expenses of poor precision, as in this case. Interesting point should have a balance in both recall and precision, which in turn will lead a good success rate.

We should compare tables 3 and 4 to state of the art models, shown as table 5. An attribute construction and selection approach appears in [5], and its finding is listed in table 5-top. The second row (from [6]) shows a model found by a stepwise forward selection method, and consists of hue, GY and  $W_r$ , which are defined as:

$$GY = -0.30 * r + 0.41 * g - 0.11 * b$$

$$W_r = \left(\frac{r}{r+g+b} - \frac{1}{3}\right)^2 + \left(\frac{g}{r+g+b} - \frac{1}{3}\right)^2$$

Unfortunately, features like  $W_r$  are very difficult to infer with the existing attribute construction scheme. Third row in table 5 shows a pragmatic and well-known approach, so called Skin Probability Map (SPM), working on raw RGB values. The SPM has a threshold variable which is tuned for this comparison. The learning procedure used all ten subsets, instead of only one. This is somehow an unfair comparison, but that is why SPMs work, see [6, 14], e.g. skin in RGB has a very sparse distribution. As a common practice in SPMs, we used the God given parameter [14] of 32 equally sized histogram bins, i.e.  $32^3$ .

color model (three components)			recall	precision	success rate
			(%)	(%)	(%)
$\frac{gb}{(r+g+b)^2}$	h	$\frac{r}{b}$	95.6	88.6	91.7
$\frac{gb}{(r+g+b)^2}$	$\frac{h*(r+g+b)}{b}$	$\frac{r}{g}$	<b>98.2</b>	77.9	85.2
$\frac{h*(r+g+b)}{b}$	$\frac{r}{g}$	h	98.1	82.9	88.9
h	$\frac{gb}{(r+g+b)^2}$	$\frac{rg}{(r+g+b)^2}$	98	78.8	85.3
h	$\frac{gb}{(r+g+b)^2}$	$\frac{b-r}{(r+g+b)}$	98	64.1	72
h	$\frac{gb}{(r+g+b)^2}$	$\frac{h*(r+g+b)}{g}$	98.1	65.3	73.4
h	$\frac{r}{b}$	$\frac{r-b}{(r+g+b)}$	94	90.5	92.4

**Table 4.** Best features generated by the attribute selection procedure. Only the last row exhibits a well-balanced performance.

other models (three components)			recall	precision	success rate
			(%)	(%)	(%)
$\frac{r}{g}$	$\frac{rb}{(r+g+b)^2}$	$\frac{rg}{(r+g+b)^2}$	93	91.5	92.2
h	GY	$W_r$	93.2	92.1	<b>92.6</b>
SPM on raw RGB			95.8	77.3	91

**Table 5.** State of the art models: (top) Automatic feature construction from RGB, (middle) Step-wise forward selection on several colour components. (bottom) Skin Probability Map, which is extremely fast.

## 6.1 Discussion

With those results on hand we may come back to the original questions. How does the hue contribute to new colour features? Essentially, there is no real impact of including hue in new features.

Although some good features have been found, we expected better results, say, by consistently exceeding a mark of 90% in both recall and precision. Two existing models, shown in table 5 do achieve this mark, and one of them does not use hue at all. Only one model in table 4 (last one, top to bottom), exhibit competitive results. And, at best, it is comparable to table 5-top, which does not use hue and thus faster to compute. Moreover, some doubts may arise with the inclusion of  $\frac{r}{b}$ , a noisy feature, which nevertheless was selected in two models.

Although promising features appear in table 3, a good enough 2D skin space is unfortunately still evasive. The success of this study relies in creating strong



features, which were not produced using hue. In fact, the attribute selection shows a strong bias to  $\frac{h*(r+g+b)}{g}$  and  $\frac{h*(r+g+b)}{b}$ . Nonetheless, their associated models show regular to poor performance.

## 7 Conclusions

We report an attribute construction experience with the aim of finding good colour features for pixel based skin detection by including hue in the initial subset. Unfortunately, with our methodology, we found that hue has a minor contribution in novel features. Moreover, only one model is at best comparable to the existing literature in this field, which does not use hue. Thus, it indicates that (i) better features may be constructed without the costly hue, and (ii) unfortunately a 2D skin colour model is still elusive.

Skin colour processing is an active field. We encourage other people to verify and extend this line by including different features (e.g. texture) in the attribute construction scheme or, perhaps, developing novel attribute selection methods that overcome the initial bias to terms with hue.

## References

1. A. Albiol, L. Torres, Edward Delp, *Optimum Color Spaces for Skin Detection*, Proc. of ICIP, Color and Multispectral Processing I, Thessaloniki, 2001.
2. J. Brand, J. S. Mason, *A comparative assessment of three approaches to pixel-level human skin-detection*, Proc. of ICPR, Vol. I, pp. 1056-1059, Barcelona, 2000.
3. J. Brand, J. S. Mason, M. Roach, M. Pawlewski, *Enhancing face detection in colour images using a skin probability map*, Int. Conf. on Intelligent Multimedia, Video and Speech Processing, pp. 344-347, Hong Kong, 2001.
4. M. Fleck, D. A. Forsyth, C. Bregler, *Finding naked people*, Proc. of ECCV, Buxton and Cipolla, eds., Vol. II, LNCS, vol. 1064, pp. 592-602, Cambridge, 1996.
5. G. Gomez, E. Morales, *Automatic Feature Construction and a Simple Rule Induction Algorithm for Skin Detection*, Proc. of the ICML Workshop on Machine Learning in Computer Vision, pp. 31-38, A. Sowmya and T. Zrimec (editors), July 2002, Sydney.
6. G. Gomez *On selecting colour components for skin detection*, Proc. of the ICPR, vol. II, pp. 961-964, Québec, Aug. 2002.
7. I. Guyon, A. Elisseeff, *An introduction to variable and feature selection*, JMLR, vol. 3, pp. 1157-1182, 2003.
8. M. Hall, H. Holmes, *Benchmarking attribute selection techniques for discrete class data mining*, to appear in IEEE Trans. on Knowledge and Data Engineering, 2003.
9. E. Hjelmås and B. K. Low, *Face detection: A survey*, CV&IU, 83(3), pp. 236-274, 2001.
10. Special Issue on Variable and Feature Selection, *Journal of Machine Learning Research*, vol. 3, Mar. 2003.
11. S. Kramer, *Demand-driven construction of structural features in ILP*, In Rouveirol and Sebag, editors, Proc. of the Int. Conf. on ILP, vol. 2157, LNAI, pp. 132-141, Springer-Verlag, Strasbourg, Sept. 2001.
12. S. Kramer, L. De Raedt, *Feature construction with version spaces for biochemical applications*, Prof. of the ICML 2001, pp. 258-265, MA, USA, 2001.

13. P. Langley, G. L. Bradshaw, H. A. Simon, *Rediscovering chemistry with the BACON system*, in Machine Learning, R. S. Michalski, J. Carbonell and T. M. Mitchell (eds.), pp. 307-329, Morgan Kaufmann, 1983.
14. M. J. Jones, J. Rehg, *Statistical color models with applications to skin detection*, Proc. of CVPR, Vol. I, pp. 274-280, CO, USA, 1999.
15. R. S. Michalski, A. Rosenfeld, Y. Aloimonos, Z. Duric, M. Maloof, Q. Zhang, *Progress On Vision Through Learning: A Collaborative Effort of George Mason University and University of Maryland*, Proc. of the Image Understanding Workshop, Palm Springs, Feb., 1996.
16. P. M. Murphy, M. J. Pazzani, *ID2-of-3: Constructive induction of M-of-N concepts for discriminators in decision trees*, Int. Workshop on Machine Learning, pp. 183-187, Illinois, June 1991.
17. F.E.B. Otero, M.M.S. Silva, A.A. Freitas, J.C. Nievola, *Genetic Programming for Attribute Construction in Data Mining*, in EuroGP 2003, C. Ryan et al. (editor), LNCS vol. 2610, pp. 384-393, Springer-Verlag, Essex, 2003.
18. G. Pagallo, *Adaptive decision tree algorithm for learning from examples*, Ph.D. thesis, University of California at Santa Cruz, 1990.
19. J.R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.
20. H. Ragavan, L. Rendell, *Lookahead feature construction for learning hard concepts*, Proc. of the ICML, pp. 252-259, Amherst, June 1993.
21. M. C. Shin, K. I. Chang, L. V. Tsap, *Does Colorspace Transformation Make Any Difference on Skin Detection?*, IEEE Workshop on Applications of Computer Vision, pp. 275-279, Orlando, Dec. 2002.
22. M. G. Smith, L. Bull, *Feature construction and selection using genetic programming and a genetic algorithm*, in EuroGP 2003, C. Ryan et al. (editor), LNCS vol. 2610, pp. 229-237, Springer-Verlag, Essex, 2003.
23. M. Soriano, B. Martinkauppi, S. Huovinenb, M. Laaksonenc, *Adaptive skin color modeling using the skin locus for selecting training pixels*, Pattern Recognition, 36(3), pp. 681-690, March 2003.
24. J.C. Terrillon, M. Shirazit, H. Fukamachi, S. Akamatsu, *Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images*, Proc. of Automatic Face and Gesture Recognition, 54-61, Grenoble, Mar. 2000.
25. N. Tsumura, N. Ojima, K. Sato, M. Shiraiishi, H. Shimizu, H. Nabeshima, S. Akazaki, K. Hori, Y. Miyake, *Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin*, ACM Trans. on Graphics, 22(3), pp. 770-779, 2003.
26. B. Zarit and B. J. Super and F. K. H. Quek, *Comparison of Five Color Models in Skin Pixel Classification*, Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 58-63, Corfu, 1999.
27. Z. Zheng, *Constructing nominal X-of-N attributes*, Proc. of the IJCAI, pp. 1064-1070, Québec, Aug. 1995.
28. Z. Zheng, *A comparison of constructing different types of new feature for decision tree learning*, in Feature Extraction, Construction and Selection: A data mining perspective, Liu and Motoda (eds.), Kluwer Academic, 1998.
29. T. Zrimec, A. Wyatt, *Learning to recognize objects - Toward automatic calibration of color vision for Sony robots*, Proc. of the ICML Workshop on Machine Learning in Computer Vision, A. Sowmya and T. Zrimec (editors), Sydney, July 2002.