Res. Lett. Inf. Math. Sci., 2005, Vol. **7**, pp 29-42
Available online at http://iims.massey.ac.nz/research/letters/

Real-time hand tracking using a set of cooperative classifiers based on Haar-like features

Andre **L**. **C**. **B**arczak[1] and **F**arhad **D**adgostar[2]

Institute of Information & Mathematical Sciences
Massey University at Albany, Auckland, New Zealand

Abstract

In this paper we discuss the importance of the choice of features in digital image object recognition. The features can be classified as invariants or non-invariants. Invariant features are robust against one or more modifications such as rotations, translations, scaling and different light (illumination) conditions. Non-invariant features are usually very sensitive to any of these modifiers. On the other hand, non-invariant features can be used even in the event of translation, scaling and rotation, but the feature choice is in some cases more important than the training method. If the feature space is adequate then the training process can be straightforward and good classifiers can be obtained.

In the last few years good algorithms have been developed relying on non-invariant features. In this article, we show how non-invariant features can cope with changes even though this requires additional computation at the detection phase. We also show preliminary results for a hand detector based on a set of cooperative Haar-like feature detectors. The results show the good potential of the method as well as the challenges to achieve real-time detection.

1 Introduction

For decades researchers in Computer Vision have tried to improve the accuracy and performance of object recognition algorithms. Yet this is still an important area of research because the algorithms are relatively primitive when comparing with mammalian vision. Computer vision development does not necessarily try to follow biological characteristics because in many circumstances it suffices to implement a simpler system that will do the specified job. However it has been increasingly useful to compare and be inspired by biological vision systems, as these have inspired the development of better techniques (Marr, 1982).

It is difficult to achieve good accuracy in image pattern recognition because the training

[1] A.L.Barczak@massey.ac.nz
[2] F.Dadgostar@massey.ac.nz

process requires a large amount of data and choosing features that characterize the object being detected is a time-consuming task. The latest trend for object recognition is the use of AI to train classifiers. For some time the focus of the work was on the training algorithms themselves. However the influence of the choice of features and the quality of the training image set quality cannot be underestimated.
(Postma, Vanden Herik, & Hudson, 1997) have studied feature based approaches and argue that there are three problems still to be solved: 1) what makes a good feature, 2) how many features should be used and 3) how to cope with the insensitivity to spatial information. In fact two of these problems are so closely relevant to this work that the discussion is presented in a different way here:
What makes a good feature – It is useful that they are invariant with respect to certain changes (specifically translation, scaling, rotation and light conditions, called *"modifiers"*) while being sensitive to the specific properties of the object in question. The problem is that there is no known generic feature that addresses all the conditions above and at the same time is easy to compute. Features that are very good at coping with modifiers may lose spatial information and vice-versa: there is a trade-off when selecting features.
How many features – There is a minimum number of features that can describe a view of an object. Too few features may not be enough to evaluate the object properly. On the other hand too many features pose a problem for training and detection phases. This problem is called "the curse of dimensionality" (Bishop, 1995). If more features are used to accurately describe an object, the training time and the required memory space is growth exponentially.
The rest of the paper discusses the feature issues as follows. The invariant features are presented briefly and their advantages and disadvantages are discused. Next the same is presented for non-invariant features. The special case of the Haar-like features is presented and the strategies to cope with translation, rotation, scaling and light changes are shown. Preliminary results regarding hand recognition are presented and discussed.

2 Invariant Features

Features associated with images are called '*invariant*' if they are not affected by certain changes regarding the object view point. It is widely accepted that invariant features would be independent of modifiers such as translation, scaling, rotation and light conditions (Wood, 1996). Ideally, invariant features should recognise objects whose geometry can change either because the object is moving in relation to the camera, is articulated or because different viewpoints cause different patterns in 2D image. Usually, these modifiers are not independent of each other and therefore they often happen simultaneously. It is also agreed that there is no truly pure invariant feature (Wood, 1996). Rather there are features that are more or less robust to one or more modifiers.
One of the simplest sets of invariant features are histograms. Histogram matching was used in the early 1990s to achieve good recognition performance under controlled conditions. However for generic computer vision applications, histograms are not necessarily good

features. All the spatial information is lost when using histograms as features. Pixels may change their positions without affecting the histogram. This problem is called the "scrambling problem" (Postma et al., 1997). In some cases it is possible to swap pixels within the image until one gets completely different objects. From the histogram's perspective an elephant may look like a fly. For an appreciation of the problem, see Figure 1. The figure shows two images produced by a web camera. Although both images have identical histograms, they clearly do not belong to the same class of objects. The same problem may occur even using colour histograms.

Despite this obvious flaw, histograms are quite useful under the right circumstances. For example, finding a ball on a robot soccer field may rely uniquely on histograms because the conditions are controlled and known in advance. A specific colour histogram can be related to the ball, as the ball has a particular colour pattern that is different than the pattern of the field and the players. The same may not be true when detecting human faces on a video sequence. An arm may present the same pattern as a face in some frames, or even wooden furniture may be considered as skin colour (if the wood colour is similar to skin colour). The specific needs of the applications may drive the choice of more specialised features.

Figure 1: Both the hand (a) and the untidy table (b) have identical histograms (c). The image (b) was produced by scrambling the pixels and changing their values slightly so its histogram would fit the other image's histogram. This simple experiment shows that the histogram does not carry enough information about the object to allow classification uniquely based on it.

Other invariant feature spaces found in the literature includes Gabor filters (Kyrki & Kamarainen, 2004), Wavelets (Schneiderman & T., 2000) and FFTs[3] (Lai, Yuen, & Feng, 2001). These features were used successfully to recognise human faces, cars, pedestrians etc. They are more robust to the scrambling problem because they consider other aspects of the image. For example, FFTs considers frequencies of parts of the image. Scrambling the image would produce different FFT results. One drawback of such features is the computation time. The computational complexity of these features is they are not typically linear. Another drawback is that there is no theory that guarantees that two different objects would not yield the same values. In other words, the features are reasonably sensitive to the

[3] Fast Fourier Transform

specific properties of the objects.
Invariant features used alone do not carry enough information for generic recognition; however they may be useful in specific circumstances and even more useful if associated with other methods. Often these features are good at some specialized recognition task and as long as the conditions are known in advance and controlled, they can help to build strong and reliable algorithms.

3 Location dependent features (non-invariant features)

The simplest non-invariant features for images would be the pixels themselves. However a set of raw pixels lacks the fundamental property of being sensitive uniquely to the object in question. The most obvious way to show this is to vary the light to obtain a completely different set of pixels regarding the same object in the same position viewed from the same viewpoint.
In his classic book, Marr (Marr, 1982) proposes features that go beyond a set of raw pixels partially based on a better understanding of the physiology of the human eye and vision strategies. Marr was one of the first to formalize the computation of edges and corners on images and the relationship of those features to object recognition. Unfortunately, the image recognition problem was much more complex than what it appeared to be, as processing edges and corners alone have not produced very robust recognition algorithms.
Edge and corner detectors have the weakness of demanding the tuning of specific parameters. Generic detectors are not efficient in dealing with images that have parts with many edges/corners together with homogeneous areas. Noisy images pose the same problem. Also, it is not always clear how to correlate specific objects with edges and corners. A more robust set of features that somehow assesses edges and corners and relates their presence to certain regions of the image would be useful. Such features exist and are called Haar-like features. Viola and Jones (P. Viola & M. Jones, 2001; P. Viola & M. Jones, 2001) were the first to develop a robust real-time algorithm based on Haar-like features. An implementation of an extended version is publicly available and described briefly in and (Lienhart, Kuranov, & Pisarevsky, 2003; Lienhart & Maydt, 2002).
Haar-like features are based on the same idea of Haar wavelets and have the following generic form:

$$f = \sum_{i=1}^{n} (w_i \times S_i)$$

where:

- w_i is the weight of a particular rectangle in a feature space
- f is the feature
- S_i represents the sum of pixels within i_{th} rectangular area in feature space (Grey-scale values between 0 to 255 are used. The feature final value is normalised to allow easier assessment at detection phase).

Typically only two or three rectangles are used in one feature. This set is rich enough to represent any object. For better use of memory and faster calculations, all the weights are small integers and relative to the areas. For example, if area 1 is three time larger than area 2, w_1 could be either 3 or -3 while w_2 could be either -1 or 1. Also the sizes of the areas are always multiples of 2, 3 or 4 and a minimum size boundary has to be met (typically at least 8 pixels). The Haar-like features can represent not only edges and lines, but also can represent subtle differences between areas in the image. We call the sub-window where the positive examples are computed at training stage, a "kernel". The Haar-like features usually represent a ratio of darkness or brightness between two or more areas within the kernel. The typical example is to state that certain regions in the human face are brighter (the cheeks) and some are darker (the eye cavities). One Haar-like feature is enough to assess such a characteristic. A whole set of them can reliably describe a face.
The size of the kernel has to be limited to 20x20 up to 50x50, otherwise training time, becomes impractically long. Special twisted features described in (Lienhart & Maydt, 2002) allow areas of pixels to be taken from 45 degrees rotated rectangles. However due to the discrete nature of pixels in a digital image it is not possible to extract areas that are twisted at any generic angle, but these additional features help to cope with small kernels, as diagonal lines and edges can be represented by them. For example suppose we need two different angles for training hands, one at 0° and one at 90°. In order to follow the same proportion of the original positive set, the first angle requires a kernel of 24x42 pixels while the second uses a 42x24 pixels kernel.
There are clear advantages of these features because at detection time they are extremely simple and fast to compute. One can compute Haar-like features efficiently using the integral image (P. Viola & M. Jones, 2001). Each feature requires only 8 or 12 operations. Very complex objects can theoretically be recognized, although the actual limitations of the technique remain an open question.
The features are also relatively robust to noise and light changes. As long as the positions of the features are kept constant in relation to the origin of the kernel it is possible to produce algorithms that are very robust to these conditions. To keep the position of the features constant, one needs to guarantee the "alignment" of the positive examples during the training phase. If the positive examples are not aligned properly then the classifier's accuracy may suffer. Assuming that the alignment problem is overcome by a careful (maybe semi-automatic) process of choosing and moving the positive images, how can these features cope with the modifiers at detection phase? The next section describes how to deal with translation, scaling, and rotation, as well as discusses the problems faced by light changes and articulated objects.

4 Alternatives for non-invariant features to deal with rotation, translation and light variations

Translation: is the simplest problem to solve using non-invariant features. The features are

computed on a fixed resolution that is smaller than the image being assessed. The kernel can sweep the image to assess the matching patterns. It takes additional computation to get the results. If the original kernel is NxM and the image resolution is W*H, the number of sub-windows is:
(W-M) * (H-N)
If there are many features to compute the additional time required to deal with translation is appreciable. Tree classifiers can help to deal with the computational effort, as not all sub-windows would have to compute the entire set of features that compose one classifier. The other problem that may arise is the fact that the classifiers can hit the same object more than once. This happens because two different sub-windows that are very close to each other can yield values that are within the margins allowed by the classifier. Usually post-processing is necessary to eliminate these additional hits and compose a single coherent hit (Figure 2).

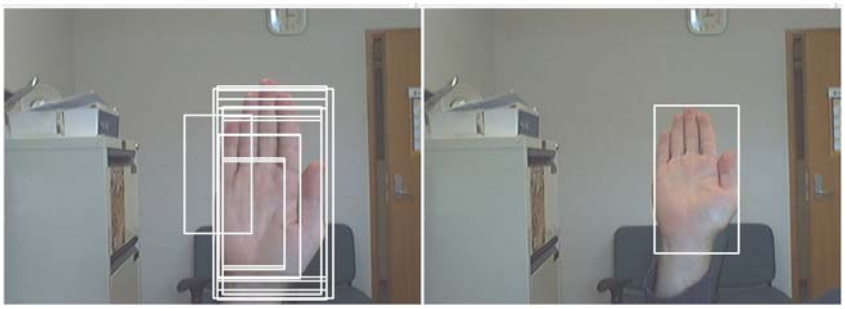

Figure 2: Left: Many hits; Right: Single hit.

***Scaling*:** Sweeping the original kernel would only find objects of the same size. Scaling is necessary to find different sizes. It would normally be a more complicated process as it involves rounding both pixel values and feature values due to the discrete nature of digital images. It is possible to compute sub-windows that are larger than the kernel by scaling them down. An easier method first proposed by Viola and Jones uses the integral image to achieve the equivalent values. Rounding problems are also overcome by their method. Once the classifier is trained, there is no need to scale down the sub-windows to be assessed, and the Haar-like features can be computed straight from the original sub-window with the help of the integral image (P. Viola & M. Jones, 2001). The integral image is computed only once for every frame.
Computing every single scale is not feasible, therefore a reasonable amount of sub-windows have to be neglected. Typically, scales are computed using factors from 1.1 up to 1.4. The smaller the factor, the more demanding the computation. If the factor is too large objects may be missed. The total number of sub-windows that has to be assessed is:

$$\sum_{i=0}^{n}(W - M \times f^{i}) \times (H - N \times f^{i})$$

where:

- W,H are the width and the height of the image
- M,N are the width and the height of the kernel
- *f* is the multiplication factor (it needs to be rounded to an integer)
- n is the maximum number of times the scaling is computed so $M.f^n < W$ and $N.f^n < H$ (See figure 3).

For example, for a 640x480 image, a kernel of size 24x42 and a factor of 1.1, the total amount of sub-windows would be 4482974. If each feature needs 8 operations and there are 15 frames per second, 4.3 x 10^8 (430365504) operations per second are needed for the calculation of the features alone. In practice translation with scaling would have to be computed in steps of more than one pixel (every *x*,y pixels) and depending on the classifier the whole image would have to be scaled down to 320x240 before anything is done, otherwise, the application can not meet the real-time requirements. The consequence is a slight loss of accuracy, so a trade-off has to be found.

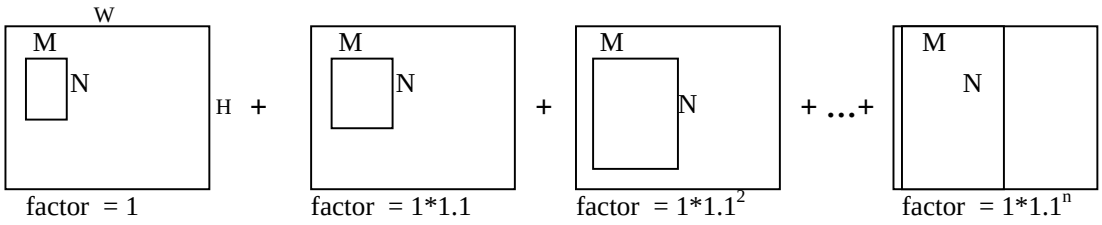

Figure 3: Scaling features to compute sub-windows with sizes that differ from the original kernel.

Rotation: is the most challenging problem. Computing rotation directly from the original sub-windows will often yield a slightly distorted image. Also the number of angles to which the comparison to the kernel has to be made can make the problem become computationally expensive. One interesting solution proposed by Rowley et al (Rowley, Baluja, & Kanade, 1998) is the use of what they called a "router" (not to be confused with the Network routers). The router is a small algorithm that computes the possible angle to which the object is twisted in relation to a fixed axis normal to the image (Figure 4). What is interesting about the router is that it can compute the angle without the knowledge that the sub-window contains an object or not. To make the concept clearer, suppose that the object is a human face. Human faces are darker in the upper half than the lower half because the cheeks are usually lighter than the eye cavities. Therefore computing which part is darker could yield a certain angle even if the sub-window does not contain a face. After the angle is known, it is the role of the classifier to decide if the sub-window contains the object or not. In Rowley's algorithm the angle is learned via NN (Neural Network) using twisted faces to train it. Rowley's results however were based on a square sub-window that is suitable for human faces. It is not clear how well the same technique would work using oblong sub-windows that are necessary to cope with objects such as hands.

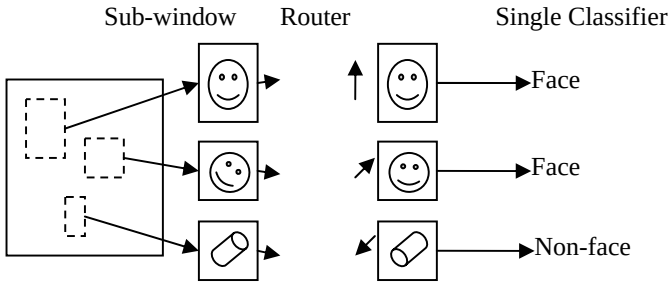

Figure 4: Rowley' router approach.

The other alternative for rotation, now specifically using Haar-like features is to train classifiers using twisted examples (Figure 5). The disadvantage is the added time and effort to train the set of classifiers, but this is compensated by the flexibility and by the control over the separate parts of this process. At detection time this multiple classifier will take more time to run. But rotating the whole image in several angles is also computationally expensive as preliminary tests have demonstrated. The work involved in rotating an image of a reasonable size (320x640 pixels) can cause the frame rate to drop significantly. Rowley's solution was implemented based on this observation. One clear advantage of using multiple classifiers is the potential for parallelisation.

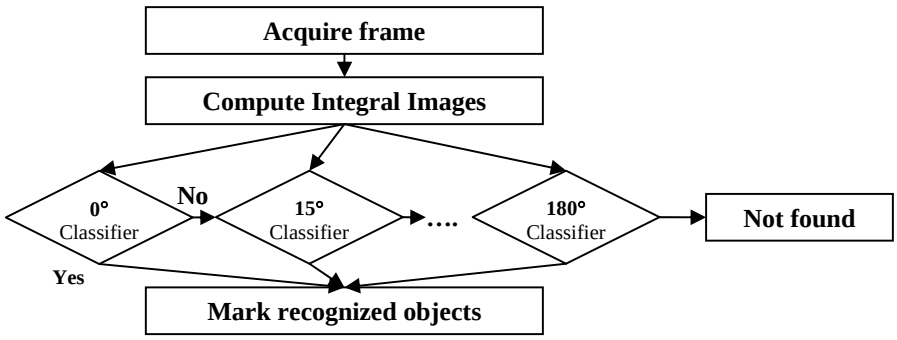

Figure 5: Rotation using many classifiers

Lighting conditions: The way an object appears may change dramatically with variations in the environment. The lighting conditions may vary not only due to sources of light and other objects producing shadows over the detectable object, but the camera itself due to automatic parameter changes that can also make the image appear lighter or darker. Haar-like features can still yield good results as long as the positive examples are representative. If new situations arise it is unlikely that the object will be detected. Therefore it is important to collect good training images because they will alone influence the accuracy of the classifier in different lighting conditions.

The current limitations of Haar-like features are with images of articulate objects such as hands. The Haar-like features are very robust as long as the object presents a stable shape. Articulate objects cannot benefit directly from this method unless many parallel classifiers are used for different shapes. Considering that many classifiers are already needed for rotation in a single viewpoint, it would be infeasible to represent all possible shapes where the articulation is too complex.
Different object proportions are not covered by the previous operations. Scaling only deals with the problem of different sizes related to the size of the image. For example figure 6 shows an exert of Picasso's Guernica. The human face and the hands can be immediately recognized by any human, but it is very difficult to make a computer vision application to generically deal with out-of-proportion objects once it is trained to recognise the standard one.
A possible solution to both problems is to have smart algorithms that could recognise separate parts and vote to decide if they compose a coherent object. In that sense when detecting faces anything with recognisable eyes, mouth and more or less round in shape could be considered as a face. When detecting hands, a number of fingers connected to a palm would be classified as a hand.

Figure 6: Different proportions can cause problems to the recognition system based on Viola and Jones

5 Experiments with Haar-like features to recognise human hands

The recognition of hands is an important step towards gesture recognition and its applications in human-computer interaction. Every gesture makes the geometry of the hand different and therefore using 2D processing algorithms, only one gesture from a single viewpoint can be classified properly; like Viola and Jones method (P. Viola & M. Jones, 2001). We have adopted one gesture (the hands in Figure 7) to assess the performance and the accuracy of the method for this application. Hand images were acquired from different people under different illumination using a dark background. An automated process segments each image to facilitate the generation of random backgrounds.

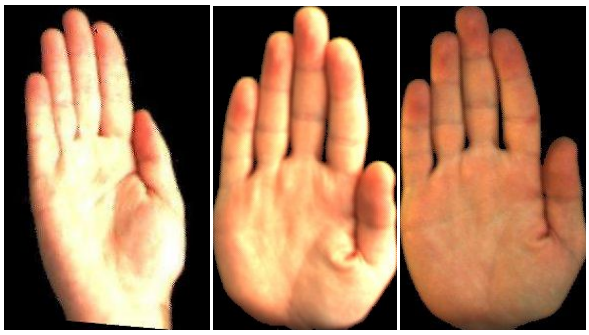

Figure 7: The basic hand images used on the training process.

In the original method proposed by Viola and Jones, translation, scaling and a certain extent of lighting conditions are already considered, but rotation is not. To experiment with hands detection we implemented a version of the method using parallel cascades. Each cascade is capable of detecting hands (one particular gesture) within a certain angle of rotation (on an axis normal to the image's plan). Some tolerance is desirable not only because it is difficult to align the positive examples accurately, but also fewer parallel classifiers are necessary.
We started the experiments setting a base of 11 hand images and using 30 different backgrounds, we made a total of 330 original positive set images that was used to train the classifier at angle 0. To automate the training process programs and scripts were used to twist the original set of images to angles from -90 to 90, spaced by 3 degrees, and a total of 61 orientations were trained. A modified version of Viola-Jones algorithm using OpenCV library was used to assess classifiers in parallel (classifier set 1). The result of the first experiment was promising, then we did the second experiment using 145 base hand images composing about 4400 images to train each classifier. The angles used in this second experiment were limited generating only 7 classifiers. Examples of the training images are shown in Figure 8 (Classifier set 2).

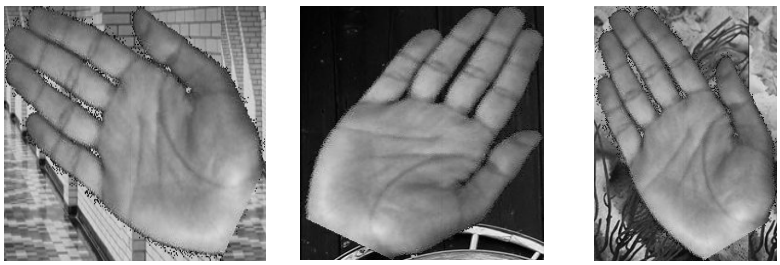

Figure 8: The positive image set example for different angles

6 Results and Discussion

OpenCV assesses different classifiers by re-computing the features of individual classifiers using the same integral images computed from each frame. The classifiers in set 1, could not learn light changes properly because 11 base images do not create enough variation. But they were capable of detecting hands. Each classifier was restricted to a small tolerance in angles and in some cases two different classifiers would hit the same hand. A simple voting system was used to eliminate any duplication. Larger angles such as 90°, classifiers did not work so well because the lighting condition in a rotated image is different than a rotated hand in real conditions.

These sets of classifiers were used to estimate the maximum possible frame rate and its relationship to the number of classifiers being used simultaneously. As expected the speed of the classification slows down due to the extra computation, but the rate drop is not linear. Currently the rate that we could achieve using a web camera on a Pentium 2.4 GHz machine with 500MB memory running Linux 2.4 is shown in Figure 9. It is interesting to observe that the frame rate drops significantly until 8 classifiers are used, and beyond this point the frame rate drops slower. This somewhat unexpected behaviour has an explanation. As the sub-windows are tested by an increasing number of classifiers, only a few will get to the bottom of the cascade while most will only be examining the first few levels. This shows a potential for exploring parallelism. Classifiers that are not 'active', i.e. are not detecting any object, will use very few resources.

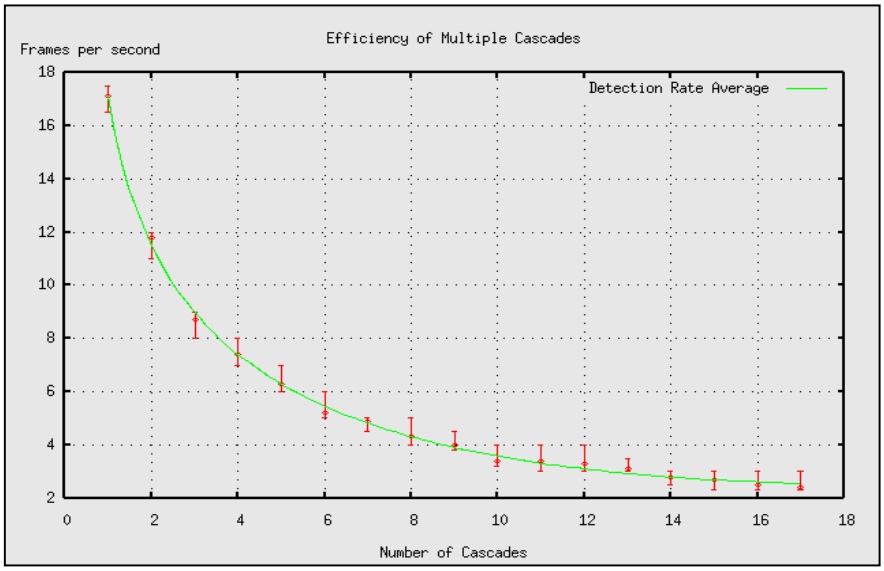

Figure 9: Rates for classifiers detecting hands in various angles.

Table 1 shows the results for classifier set 2. The primary results show an average hit rate

of 62.7% hits, with a maximum of 84% and minimum of 34% detection. For larger angles the results are very poor; indicating the base images did not have enough variations. False positive rate on average was around 7.7% which is considered good.
The results show that using non-invariant features such as Haar-like features can produce relatively reliable detectors. Hands are difficult because any small change in geometry causes detection failure. So far the results, although very limited, but are promising. Location dependent features can overcome a few of the problems related to the misalignment of the positive examples. However the patterns that include shadows and the partial occlusion can only be learned if the examples include these situations.
We should note that, Haar-like features chosen for all angles, does not bring acceptable results. The reason seems to be the rounding problem. Figure 10 shows a comparison of three angles and shows that it is not possible to have the same features, as the discrete nature of the pixels and the fact that they are defined as a square region do not allow twisting features at any angle.

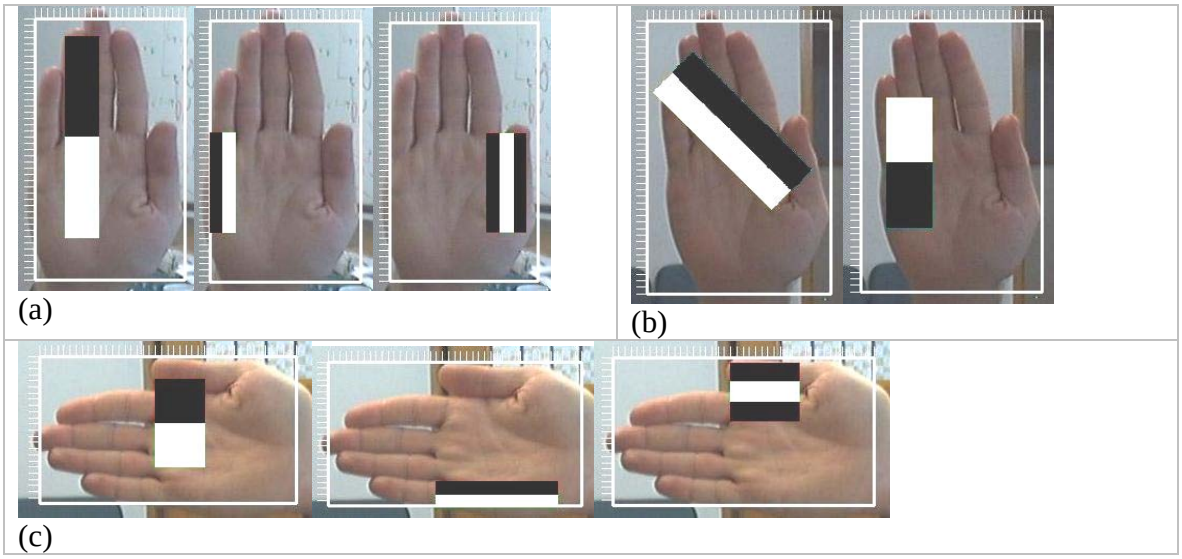

Figure 10: Although the same examples were used, features are not equivalent when rotating the examples and retraining. a) The first set of features for 0°. b) The first set of features -15° c) The first set of features to -84°.

The limitations of non-invariant features are related to situations where the object appears very differently due to view points, or they present a very different colour pattern (a car is a typical example). This can be overcome by either training a multi-level classifier or by training parallel classifiers to deal with the various types of images that are associated with the same object.

Table 1: Hit-ratio and false-positives for the first test set

Angle	Number of Samples	Hits	False Positives	%Hit	%False Positives
-90	109	37	7	34	6.4
-60	104	65	9	62.5	8.6
-30	117	92	10	78.6	8.5
0	100	72	10	72	10
30	121	85	17	70.2	14
60	100	84	4	84	4
90	129	63	3	48.8	2.3
Total	780	498	60	62.7	7.7

7 Conclusions and Future Work

Invariant features would have the advantage of dealing with translation, scaling, rotation and lighting changes, but often they lack of the important property of uniquely correlating to a certain object. Performance and accuracy of classifiers using non-invariant features such as Haar-like features can be stretched by a careful design of a recognition algorithm that allows these operations to be carried out. While translation and scaling are straightforward problems, rotation and light condition are not. Rotation demands a lot of extra computation which can be overcome by Rowley's idea (Rowley et al., 1998) of routers. Lighting condition accuracy is totally dependent on the set of positive examples. If this set is rich enough to cover a wide variety of situations the classifier is very robust.
A combination of both feature types would be desirable in some cases, but how to achieve the best for generic recognition is still an open problem.
In future work we intend to carry out more experiments to improve the performance and the accuracy of hand recognition. These include a more efficient rotation algorithm, a pre-processing histogram based on skin colour and using motion features of the image.

Acknowledgements

The authors would like to thank Dr. Chris Messom, Dr. Hossein Sarrafzadeh and Dr. Martin Johnson for their valuable suggestions and support for the publication of this paper. We also acknowledge the use of IIMS parallel computer known as the *Sisters* for most of the training computation that produced the classifiers for the preliminary results. .

References

Bishop, C. M. (1995). *Neural network for pattern recognition*: Oxford University press.

Kyrki, V., & Kamarainen, J. (2004). Simple Gabor Feature Space for Invariant Object

Recognition. *Pattern Recognition Letters, 25*(3), 311-318.

Lai, H. J., Yuen, P. C., & Feng, G. C. (2001). Face recognition using holistic Fourrier invariant features. *Pattern Recognition Letters, 34*, 95-109.

Lienhart, R., Kuranov, A., & Pisarevsky, V. (2003). *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection.* Paper presented at the 25th Pattern Recognition Symposium DAGM 2003, Madgeburg, Germany.

Lienhart, R., & Maydt, J. (2002). *An Extended Set of Haar-like Features for Rapid Object Detection.* Paper presented at the IEEE ICIP 2002.

Marr, D. (1982). *Vision*. San Francisco: W.H. Freeman.

Postma, E., Vanden Herik, J., & Hudson, P. (1997). Image Recognition by Brains and Machines, *Brain-like Computing and Intelligent Information Systems*: Springer Verlag.

Rowley, H., Baluja, S., & Kanade, T. (1998). *Rotation Invariant Neural Network-Based Face Detection.* Paper presented at the Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Schneiderman, H., & T., K. (2000). *A Statistical Method for 3D Object Detection Applied to Faces and Cars.* Paper presented at the Conference on Computer Vision and Pattern Recognition.

Viola, P., & Jones, M. (2001). *Rapid Object Detection Using a Boosted Cascade of Simple Features.* Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).

Viola, P., & Jones, M. (2001). *Robust real-time object detection.* Paper presented at the Second International Workshop on Theories of Visual Modelling Learning, Computing, and Sampling.

Wood, J. (1996). Invariant pattern recognition: a review. *29*(1), 1-17.