Fast and Efficient Rotated Haar-like Features using Rotated Integral Images

Chris Messom and Andre Barczak
Institute of Information and Mathematical Sciences
Massey University, New Zealand
C.H.Messom@massey.ac.nz

Abstract

This paper introduces an extended set of Haar-like features beyond the standard vertically and horizontally aligned Haar-like features [Viola and Jones, 2001a; 2001b] and the 45^o twisted Haar-like features [Lienhart and Maydt, 2002; Lienhart *et al.*, 2003a; 2003b]. The extended rotated Haar-like features are based on the standard Haar-like features that have been rotated based on whole integer pixel based rotations. These rotated feature values can also be calculated using rotated integral images which means that they can be fast and efficiently calculated with just 8 operations irrespective of the feature size. In general each feature requires another 8 operations based on an identity integral image so that appropriate scaling corrections can be applied. These scaling corrections are needed due to the rounding errors associated with scaling the features. The errors introduced by these rotated features on natural images are small enough to allow rotated classifiers to be implemented using a classifier trained on only vertically aligned images. This is a significant improvement in training time for a classifier that is invariant to the rotations represented in the parallel classifier.

1 Introduction

Safe and reliable human-robot interaction needs a vision system that can robustly identify people, recognise them and track there body parts as well as recognise any gestures executed by that person.

Haar-like features have been used successfully in face tracking and classification problems [Lai *et al.*, 2001; Jones and Viola, 2003; Barreto *et al.*, 2004; Huang and Lai, 2004], however other problems such as hand tracking [Barczak *et al.*, 2005; Micilotta and Bowden, 2004; Kölsch and Turk, 2004] have not been so successful. The main reason for this is the fact that Haar-like features are not invariant over rotation. This means that any object that rotates and is sensitive to angle changes (such as hands) will be difficult to solve using standard Haar-like features. The features that define faces tend to be insensitive to small angle variations and Haar-like features have been used to detect head rotations of as much as 15^o from the vertical [Jones and Viola, 2003]. When people are standing their head is naturally aligned vertically with respect to gravity and so this rotational sensitivity tends not to be a significant problem for faces. Other body parts such as hands, arms and legs are not normally alligned with the horizontal or vertical axes so are difficult to model with traditional Haar-like features. Researchers have tended to use edge detection or colour based tracking of these parts [Messom *et al.*, 2006].

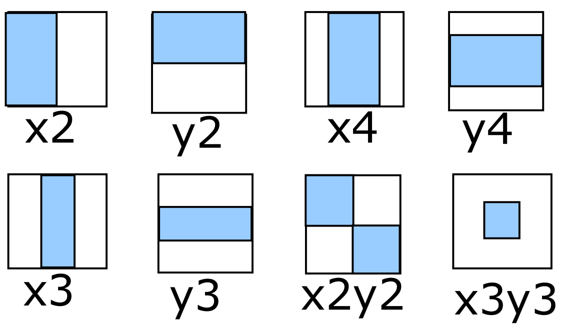


Figure 1. Standard Haar-like features.

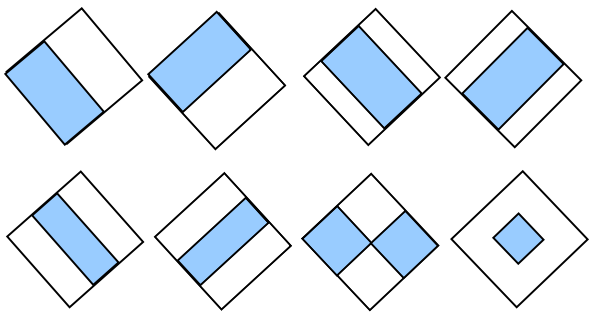


Figure 2. 45^o twisted Haar-like features.

Several researchers have studied the impact of in plane rotations with the use of twisted Haar-like feature (45^o) [Lienhart and Maydt, 2002; Lienhart *et al.*, 2003a; 2003b] or diagonal features [Viola and Jones, 2001b] fairly good performance has been achieved. These techniques will have little benefit for problems that are sensitive to rotations, such as hand identification [Barczak *et al.*, 2005; Kölsch and Turk, 2004; Antón-Canalís et al., 2006; Stenger et al, 2004; Wachs et al., 2005] which are not aligned to fixed angles (0^o, 45^o, 90^oetc).

2 Standard Haar-Like Features

Standard Haar-like features consist of a class of local features that are calculated by subtracting the sum of a subregion of the feature from the sum of the remaining region of the feature. This is illustrated by figure 1. These feature are characterised by the fact that they are easy to calculate and with the use of an integral image, very efficient to calculate.Lienhart *et al.* [2002] introduced an extended set of twisted Haar-like feature, illustrated in figure 2. These are the standard Haar-like feature that have been twisted by 45^o. Lienhart *et al.* [2002] did not originally make use of the twisted checker board Haar-like feature (x2y2) since the diagonal elements that they represent can be simply represented using twisted features, however it is clear that a twisted version of this feature can also be implemented and used.

These twisted Haar-like features can also be fast and efficiently calculated using an integral image that has been twisted 45^o. The only implementation issue is that the twisted features must be rounded to integer values so that they are aligned with pixel boundaries. This process is similar to the rounding used when scaling a Haar-like feature for larger or smaller windows, however one difference is that for a 45^o twisted feature, the integer number of pixels used for the height and width of the feature mean that the diagonal coordinates of the pixel will be always on the same diagonal set of pixels, see figure 3. This means that the number of different sized 45^o twisted features available is significantly reduced as compared to the standard vertically and horizontally aligned features.

3 Integer Rotated Haar-LikeFeatures

General rotations of Haar-like features can not be easily implemented efficiently, therefore we define a restricted set of rotations called integer rotations that can be easily and efficiently implemented.

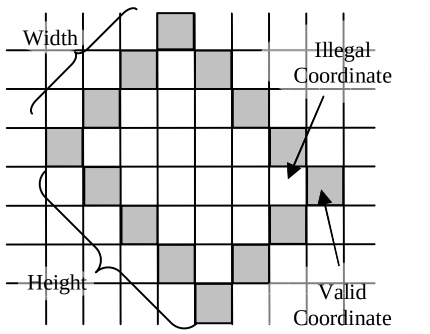


Figure 3. Opposite Corners of 45^o twisted Haar-like feature on identical diagonal.

An integer rotated Haar-like feature is a feature that has been rotated by an angle arctan(A/B) where A and B are integers. This means that an integer rotated line consists of all angles that have a rational tangent. A 45^o rotated Haar-like feature is a special case of a feature which has been 1-1 integer rotated.

A unit-integer rotated Haar-like feature is a feature that has been rotated by an angle arctan(A/B) where A and B are integers and either A or B is 1. A 45^o rotated Haar-like feature is a special case of a feature which has been 1-1 unit-integer rotated. Figure 4 illustrates a set of 1-2 rotated Haar-like features.

This paper will discuss unit-integer rotated features, which restricts the angles available to those listed in table 1. The table shows that a large number of unit-integer rotations are available near the horizontal or vertical while only a few are available near 45^o. In practise depending on the coverage required, a selection of these unit-integer rotations will be chosen, for example if rotations of about 10^o to 20^o degree increments are needed for a particular problem then 1-1, 1-2, 2-1, 1-4 and 4-1 will be used as well as the standard horizontal and vertically aligned features giving 0^o, 14^o, 26.5^o, 45^o, 63.5^o, 76^o, 90^o. The rotations in the other three quadrants are given by simple reflections in the x and y axes. If a higher precision and accuracy are required, rotations of less than 10^o would need a non unit-integer rotations such as 2-3 and 3-2 rotations giving angles of 37^o and 53^o.

The availability of rotated features means that a fully trained classifier using the standard features can be transformed to a rotated version. For example a face tracking system that is reliable for vertically aligned faces within a range of $\pm 20^o$ will be rotated so that it can classify faces aligned within $\pm 20^o$ of any of the unit-integer rotated axes such as 45^o $\pm 20^o$, 26.5^o $\pm 20^o$ etc, effectively producing a parallel classifier (similar to that of Rowley [1998] that can cover all possible rotations.

4 Implementation Issues for Integer Rotate Haar-Like Features

When a feature is rotated the position of the top left corner of the feature will be defined by the rotation angle and the position of the feature in the kernel. The height and width of the feature then determines which pixels form the feature itself. Rounding will cause the height and width of the feature to be aligned with the pixel boundaries.

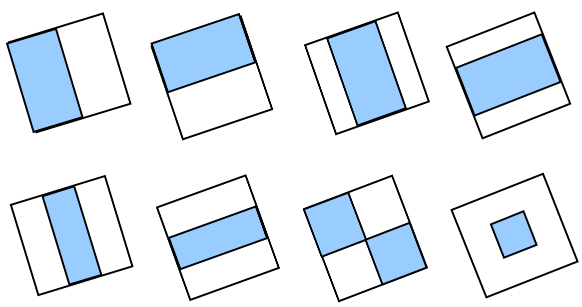


Figure 4. 1-2 Rotated (26.57^o) Haar-like features.

Table 1. Unit-Integer rotation angles.

[redacted]			
1-1	45^o	1-1	45^o
2-1	63.43^o	1-2	26.57^o
3-1	71.57^o	1-3	18.43^o
4-1	75.96^o	1-4	14.04^o
5-1	78.69^o	1-5	11.31^o
6-1	80.54^o	1-6	9.46^o
7-1	81.87^o	1-7	8.13^o
8-1	82.87^o	1-8	7.13^o
9-1	83.66^o	1-9	6.34^o
10-1	84.29^o	1-10	5.71^o
etc	..	etc	..

Since the pixelation of raster lines are not unique we need to choose a standard rasterisation so that the features provide consistent values. In this paper we rasterise based on the position of the starting point of the feature in the image.

A standard rasterisation is chosen so that a change in horizontal pixel position occurs every n vertical pixels for 1-n rotated vertical lines, while a change in vertical pixel position occurs every n horizontal pixels for 1-n rotated horizontal lines. This is calculated using equations 1 and illustrated in figure 5.

$$\alpha(x,y) = O(x,y)+[w*\cos\theta, -(w*\cos\theta)/n-((w*\cos\theta)/n+x) \bmod n]$$
$$\beta(x,y) = O(x,y)+[(h*\sin\theta)/n + ((h*\sin\theta)/n+y) \bmod n, h*\sin\theta]$$
$$\gamma(x,y) = H(\beta(x,y)) \cap V(\alpha(x,y)) \qquad (1)$$

where O(x,y) are the coordinates of the start of the feature, $\alpha(x,y)$, $\beta(x,y)$ and $\gamma(x,y)$ are the calculated coordinates of the top right, bottom left and bottom right of the feature based on the height h and width w of the feature, θ is the 1-n unit-integer rotation angle, $H(\varepsilon)$ and $V(\varepsilon)$, represents the "horizontal" and "vertical" lines through ε in the unit-integer rotated image, $\cap$ represents the intersection operator of two lines, * is integer multiplication and mod is the integer modulo operator.

Figure 6a shows a feature of height 5 and width 4 that has been 1-2 rotated. The figure also shows that a general unit-integer rotation (other than 1-1 rotations) in a digital image result in raster lines that are pixelated.

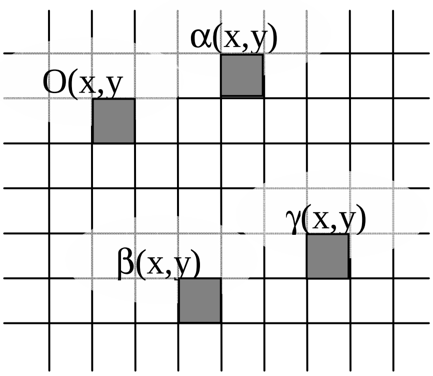
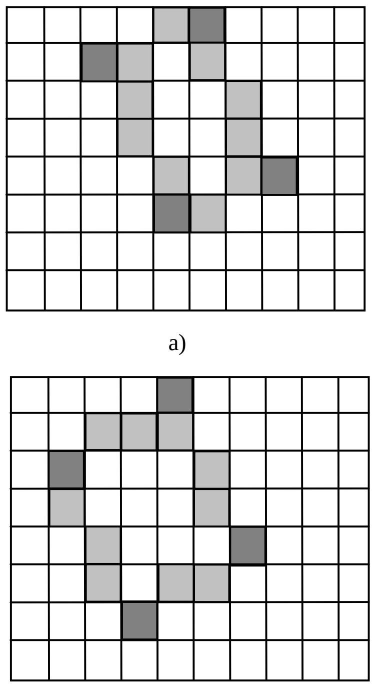


Figure 5. Calculation of feature coordinates in Integral Image.

The pixelation between two points is in general not unique as this depends on the position in which the line begins as illustrated by figure 6b. Large feature sizes as compared to the integer rotation size (n for 1-n and n-1 integer rotations) will result in similar raster lines, but will still provide different feature values when evaluated as they consist of different pixels. Using a standard rasterization a single integral image can be used to evaluate the feature, whatever the size and starting position of the feature. For some values of h and w and and angle θ the lines ($\beta(x,y)$) and $V(\alpha(x,y))$ will not interect. To overcome this problem, h, w or both h and w may need to be modified ($\pm$1 pixel), so that the do intersect. This is the similar problem to the 45^o twisted features where h and w must be modified so that opposite corners are on a valid coordinate (see figure 3).

4.1 Rotated Integral Image

The rotated integral image for a given integer rotation is calculated by summing the pixels in the relevant aligned quadrant above the given pixel (see figure 7). For the normal integral image with no rotation this is the quadrant above and to the left of the current pixel (figure 7a) . Buffering the previous row column sums allows this integral image to be efficiently implemented [Viola and Jones, 2001a]. [Lienhart and Maydt, 2002] extended this algorithm to the twisted integral image (which is equivalent to the 45^o rotated integral image, or the 1-1 integer rotated image). This twisted integral image was efficiently implemented by providing buffers of the pixels along diagonal rows.



a)

b)

Figure 6. a. A 1-2 rotated feature of height 5 and width 4. b. A 1-2 rotated feature of height 5 and width 4 starting from a new position.

The algorithm to calculate the integer rotated integral images is also efficiently implemented by providing rotated column sum buffers of the previous rows pixels. The implementation of these buffers are more complex than the twisted or normal buffers which contributes to the slower calculation of the rotated integral image as compared to the twisted or normal integral images. For simplicity, rotated integral images that are greater than 45^o are processed left to right while rotated integral image that are less than 45^o are processed right to left.

This is done because the only quadrant that depends only on the pixels above the current point in the integral image is the "top right" quadrant for angles less that 45^o (figure 7c) and the "top left" quadrant for angles greater that 45^o (figure 7c). When the angle of rotation is less than 45^o the line of pixels that have to be summed are the pixels from the current point to the end of the line. Processing right to left essentially just mirrors the left to right processing of the algorithm for greater than 45^o rotated integral images. An alternative approach of processing columns rather than rows is inefficient as the image must be completely in memory before processing can begin.

4.2 Scaling correction factors

When the Haar features are scaled for use in a classifier the feature values must be corrected based on distortions due to uneven scaling of the different areas of the features. This is achieved by using an identity integral image to count the number of pixels in each feature area and scaling the areas appropriately. The corrected feature values are given by equation 2.

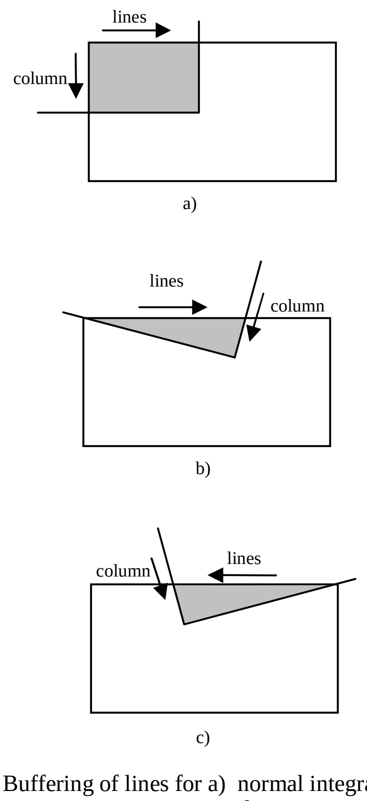


Figure 7. Buffering of lines for a) normal integral image, b) $>45^o$ rotated integral image c) $<45^o$ rotated integral image.

$$F = [A_\alpha * f_\beta / A_\beta] - f_\alpha \quad (2)$$

where A_α and A_β are the areas of the feature and the sub-features respectively (obtained from a rotated identity integral image) and f_α and f_β are the values of the sum of the pixels in the feature and sub-features respectively (obtained from a rotated integral image).

The time and space complexity of the algorithm to calculate the rotated integral image depends on the size of the original image being processed. If this image is small enough so that significant portions can fit in cache, the performance of the algorithm can be significantly improved. Therefore the rotated integral image performance is compared to that of the normal integral image algorithm, this relative performance will give an indication of the potential usefulness of this approach.

Table 2 was compiled from running the normal, twisted and rotated integral image algorithm on an opteron 250 processor (2.4GHz with 1MB cache). It can be seen from table 2 (shaded columns) that over a range of image sizes the integer-rotated integral images are slower to calculate than the normal integral images by an order of magnitude (between 8.5 and 14 times slower), but they are not significantly slower to calculate than the twisted integral image. These results show that the integer-rotated integral image can be used in real time image processing systems.

The rotated Haar-like features require just 4 operations per subfeature or 8 operations per feature independent of size, similar to the normal and twisted Haar-like features. One difference in comparison to the efficient Haar-like features that have been implemented with OpenCV is that rounding for rotated feature sizes are dependant on the starting position of the feature due to the raster effect of the rotated integral image. This means that the naïve equivalence of the time complexity will not be the case in practice. This is to account for the repeated calculation of the rounded feature positions based on the starting point of the feature.

5 Evaluation

The rotated features introduced in this paper can be used in classifiers to identify rotated objects in the image. To test the accuracy of the rotated features they were evaluated on natural images. The images were rotated between 0^o and 90^o (rotated every 1^o) and then the feature values were compared to the equivalent feature in the equivalent position in the original image. Figure 8 shows a representative set of rotated images that were tested. The images were artificially rotated. Naturally rotating the objects would be a laborious process that requires the matching positions in the image to be acurrately recorded so that the corresponding feature values could be compared. The human error in using naturally rotated object images would further complicate the analysis of the accuracy of the rotated features.

Table 2. Time complexity of Integral Image calculation including ratio of twisted and rotated features versus normal features

Image Size	Normal/ ms	Twisted/ ms	Twisted/ Normal	Rotated/ ms	Rotated/ Normal
320x240	0.3	4.2	14	4.2	14
640x480	1.6	4.4	2.75	17.2	10.75
1280 x 960	8	22	2.75	68	8.5

Figure 9 shows the average error of a unit-integer rotated feature over the range of angles from 0^o to 90^o. The minimum average error corresponds to the unit-integer angle for that particular feature and this error is very small as compared to the maximum possible feature values. The maximum errors for each feature at the unit-integer angle is small and only occurs at various degenerative cases that do not often occur in natural images and often do not correspond to positions in an image that match a given classifier. This means that the unit-integer rotated features can be successfully used in a classifier.

Parallel classifiers for 1-1, 1-2 and 2-1 unit-integer rotated features were tested and provided accurate classification, although not as good as the original vertically and horizontally aligned classifier. These rotated classifiers were derived from a classifier using standard features and vertically aligned faces. Performance may be improved by actually training classifiers using the rotated features themselves, but for a large number of rotations, this would significantly increase the training time.

6 Conclusions

This paper has presented novel rotated Haar-like features that can be used to produce rotationally invariant classifiers. The unit-integer rotated integral image can be calculated at a speed similar to twisted features, with similar constant time calculation of feature values, independent of feature size. The accuracy of the rotated features were compared to standard features and it was shown that the average and maximum errors (for natural images) are constrained allowing the features to be successfully used in a parallel classifier (figure 10). The time required to execute the parallel classifier (with n unit-integer rotated classifiers) is significantly less than n times the time to execute a normal classifier as the longest processing occurs when the image contains a positive classification example, which only occurs in at most one of the unit-integer classifiers. The parallel rotated classifiers are only trained on aligned data and then are converted into equivalent rotated classifiers, so reducing the training time as compared to a parallel classifier that is trained on rotated data.

Future work focusses on automating the tuning of the rotated parallel classifiers derived from the original vertically and horizontally aligned classifiers, so that performance can be improved to match that of the original classifiers. Classifiers that include all the features in a single classifier will also be investigated for identifying objects that have multiple angles in a single image (such as an open hand).

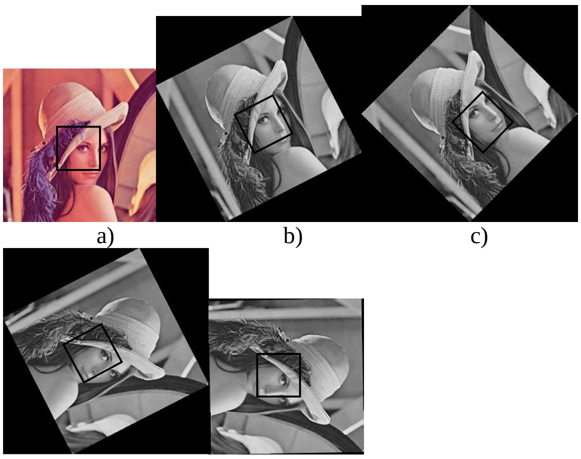
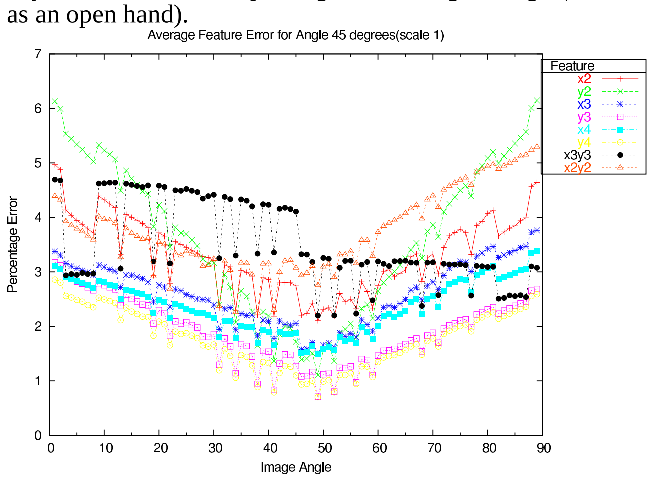
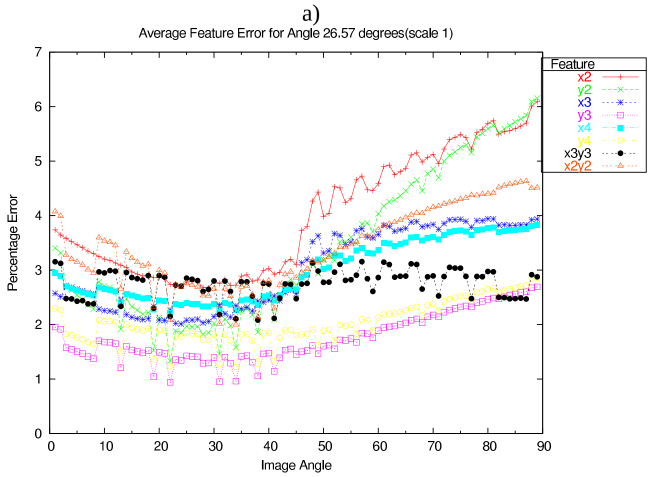


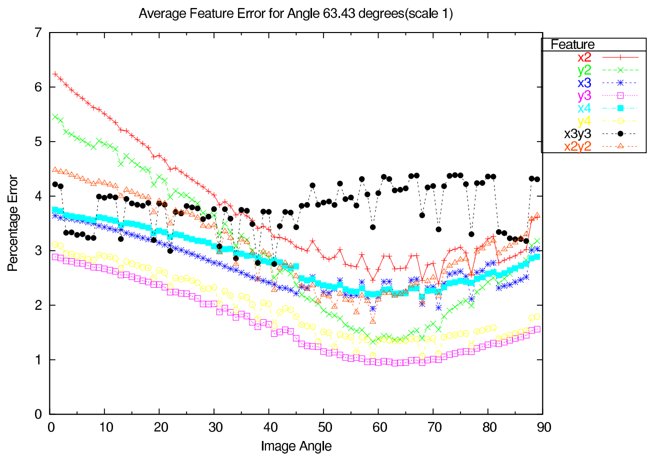
Figure 8. a) Original image b) 27^o rotated image c) 45^o rotated image d) 63^o rotated image e) 89^o rotated image and an example equivalent feature boundary.



a)



b)



c)

Figure 9. Average errors for Unit-Integer Rotated features, a) 45^o rotated features b) 26.57^o rotated features c) 63.43^o rotated features

References

[Antón-Canalís et al., 2006] L. Antón-Canalís, E. Sánchez-Nielsen and M. Castrillón-Santana, Fast and Accurate Hand Pose Detection for Human-Robot Interaction, *Pattern Recognition and Image Analysis*, Springer-Verlag, Berlin, LNCS 3522, pp. 553–560, 2005

[Barczak et al., 2005] A.L.C. Barczak, F. Dadgostar and C.H. Messom, Real-Time Hand tracking based on non-invarient features, *IEEE Instrumentation and Measurement Technology Conference*, Ottawa, Canada, May 2005, pp 2192- 2197, ISBN 0-7803-8879-8.

[Barreto et al., 2004] J. Barreto, P. Menezes, and J. Dias, Human-robot interaction based on haar-like features and eigenfaces, *International Conference on Robotics and Automation*, New Orleans, 2004.

[Huang and Lai, 2004] Szu-Hao Huang and Shang-Hong Lai, Detecting Faces from Color Video by Using Paired Wavelet Features, *2004 Conference on Computer Vision and Pattern Recognition Workshop*, Volume 5, 2004, Washington, D.C., USA.

[Lai et al., 2001] Lai, H. J, Yuen, P. C. and Feng, G. C., Face recognition using holistic Fourrier invariant features, *Pattern Recognition*, v. 34, pp 95-109, 2001.

[Lienhart and Maydt, 2002] R. Lienhart and J. Maydt, An Extended Set of Haar-like Features for Rapid Object Detection, *IEEE ICIP 2002*, Vol. 1, pp. 900-903, Sep. 2002.

[Lienhart et al., 2003a] R. Lienhart, A. Kuranov and V. Pisarevsky, Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, *DAGM'03, 25th Pattern Recognition Symposium*, Madgeburg, Germany, pp. 297-304, Sep. 2003.

[Lienhart et al., 2003b] R. Lienhart, L. Liang and A. Kuranov, A Detector Tree of Boosted Classifiers for Real-time Object Detection and Tracking, *IEEE ICME2003*, Vol. 2, pp. 277-280, July 2003.

[Jones and Viola, 2003] M. Jones and P. Viola, *Fast Multi-view Face Detection*, Mitsubishi Electric Research Laborotories, TR2003-96 July 2003.

[Kölsch and Turk, 2004] M. Kölsch, M. Turk., Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector, *IAPR International Conference on Pattern Recognition*, 2004.

[Micilotta and Bowden, 2004] A. Micilotta and R. Bowden, View-based Location and Tracking of Body Parts for Visual Interaction, *BMVC 2004*, Kingston, 7th-9th Sept, 2004.

[Messom et al., 2006] C.H. Messom,G. Sen Gupta and S. Demidenko, "Hough Transform Run Length Encoding for Real-time Image Processing", *IEEE Transactions on Instrumentation and Measurement*, to appear, 2006.

[Rowley et al., 1998] H. Rowley, S. Baluja and T. Kanade, Rotation Invariant Neural Network-Based Face Detection, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June, 1998.

[Stenger et al, 2004] B. Stenger, A. Thayananthan, P. Torr and R. Cipolla, Hand Pose Estimation using Hierarchical Detection. *ECCV Workshop on HCI 2004*, Lecture Notes in Computer Science, Springer-Verlag, vol. 3058, pp. 102-112.

[Viola and Jones, 2001a] P. Viola and M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Vol. 1, pp. 511-518, December 2001.

[Viola and Jones, 2001b] P. Viola and M. Jones, Robust real-time object detection, *Second International Workshop on Theories of Visual Modelling, Learning, Computing, and Sampling*, 2001.

[Wachs et al., 2005] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith and J. Handler, A Real-Time Hand Gesture System based on Evolutionary Search, *Genetic and Evolutionary Computation Conference*, GECCO 2005.

Figure 10. (Top) Classifier based on standard features in a Viola-Jones classier fails when given large (45^o) in plane rotations. (Bottom) Using two rotated Viola-Jones classifiers (shown using orange and red) and one standard Viola-Jones classifier (shown using yellow) the face can be identified both with and without in plane rotations. Note: as shown in the bottom left image some spurious face candidates are intermittently identified, some of these can be fitered out using the candidate faces from a sequence of images in a video.