

## Real-time Computation of Haar-like features at generic angles for detection algorithms

A. L. C. BARCZAK, M. J. JOHNSON & C. H. MESSOM

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

This paper proposes a new approach to detect rotated objects at distinct angles using the Viola-Jones detector. The use of additional Integral Images makes an approximation the Haar-like features for any given angle. The proposed approach uses different types of Haar-like features, including features that compute areas at  $45^\circ$ ,  $26.5^\circ$  and  $63.5^\circ$  of rotation. Given a trained classifier (using normal features) a conversion is made using a pair of features so an equivalent value is computed for any angle. This conversion is only an approximation, but the errors are constrained and they would have limited impact on the final accuracy of the classifier. We discuss the sources of errors in the computation of the Haar-like features and show through experiments that in natural images the errors are often negligible.

**Keywords:** Pattern Recognition, Rotation Invariant Detectors, Real-time Object Detection, Haar-like Features, Viola-Jones Detector.

---

### 1 Introduction

The Viola-Jones detector (1)(2) has received considerable attention since its publication. It has been used mainly for face detection (3), face recognition (4) and hands detection (5). Other uses include robot-soccer ball detection(6) and ecological applications such as wild life surveillance(7).

Due to the non-invariant nature of the Haar-like features, classifiers trained with this method are often incapable of finding rotated objects. It is possible to use rotated positive examples during training, but such a monolithic approach often results in inaccurate classifiers (8). Besides such a classifier would not inform at which angle the object was found. Alternatively one can train a set of classifiers, each one specialised in a certain angle interval. Although this might achieve the detection accuracy required, it also makes the training computationally much more expensive. Ideally one should be able to train a classifier once and detect the object at a generic angles.

The Viola-Jones detector compute Haar-like features using Integral Images, a special data structure that speeds-up the calculation. In practise it is only possible to compute Haar-like features accurately at a fixed angle. The angle is determined by the way the Integral Image is created. Currently two types of Integral Images are in use, one capable of computing areas at  $0^\circ$  and another at  $45^\circ$ . If the feature type is converted it is also possible to compute features at other quadrants ( $90^\circ$ ,  $-90^\circ$  and  $180^\circ$  from the original) using the same Integral Image (8). Therefore one can compute features at eight different angles using two Integral Images.

The solution proposed by this paper makes use of two additional Integral Images that computes angles at  $26.5^\circ$  and  $63.5^\circ$  to approximate the Haar-like feature values at a generic angle. We show that this approximation is good enough for detection by analysing the errors associated with such approach. This alternative for rotation keeps the training time constrained to that of a single classifier and has only moderate impact on the detection runtime.

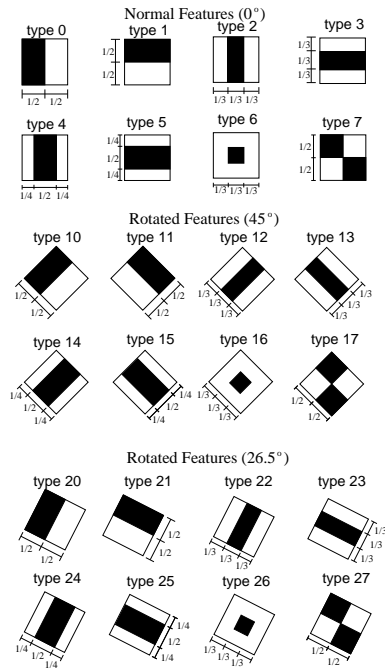


Figure 1: Classifiers using the Normal features can be converted to the other types.

The paper is organised as follows: a brief description of the Viola-Jones methods and algorithms are presented (training and detection), including some important extensions added by other authors. Next the proposed method for detection with rotation is explained. The following section presents experiments regarding error analysis. Finally the conclusions point out the limitations and possible improvements on a generic rotation invariant detector using Haar-like features.

## 2 Viola-Jones Detector

The Viola-Jones detector has three main characteristics: it uses an over-complete set of Haar-like features, it uses Integral Images to compute areas (sum of pixels) in a very efficient way and it uses Adaboost for training.

Haar-like features can be constructed in many shapes and computed in different ways. Figure 1 shows three groups of Haar-like features. The original implementation (1) only used types 0,1,2,3 and 7. The implementation in OpenCV (9) used all types but types 7 and 17. The third group was implemented for the error analysis presented in this paper. The details of the Integral Image concept can be found in (1).

Viola and Jones used a customised version of Adaboost, which was first created by Freund and Schapire (10) to solve machine learning problems. One of the changes made to the algorithm was the creation of many layers. Each layer is trained by several rounds of Adaboost. To improve training speed as well as detection performance they introduced some heuristics.

Since the time the Viola-Jones detector was first published many variations have been proposed, of which we cite three. The first is the empirical analysis carried out by Lienhart et. al. (11), where they compared three Adaboost algorithms called Discrete, Real and Gentle. Their experiments, which were limited to face detection, pointed to the Gentle Adaboost as the more accurate method. The second is the introduction of the Floatboost algorithm by Li et. al. (12). Floatboost allowed them to create classifiers with a smaller error margin with fewer features per layer. The third is

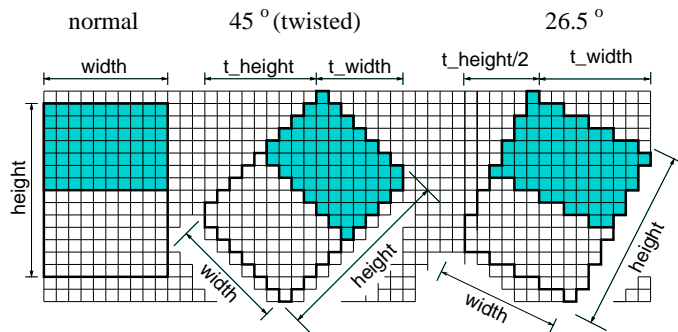


Figure 2: An approximated equivalence between a normal and a  $26.5^\circ$  Haar-like feature.

the introduction of a fast heuristic to find a sub-optimal feature by McCane and Novins (13). They reported an improvement in the training time with just minor effects on the detection phase.

## 2.1 Error sources in the computation of Haar-like features

In practise when computing the Haar-like features in digital images one cannot expect an exact value. Although the features are a simple comparison between areas (sum of pixels), there are three main sources of errors:

- the rounding up of the feature sizes due to scale changes
- the rounding up of the position of the feature in relation to a fixed point in the image
- the approximation due to some pre-processing of the image (rotation, scaling etc).

The first error source can be partially compensated by a correction factor that ensures that the areas are proportional to the theoretical definition of that particular feature. Lienhart et al. (11) suggested a correction to this problem. A correction factor is computed so that the weights of the different rectangles of a feature keep the original area ratio between them. One experiment that should be done when testing any implementation is to find the value of the features in a grayish image with equal pixel values all over the image. All features at any size and scale should yield zero.

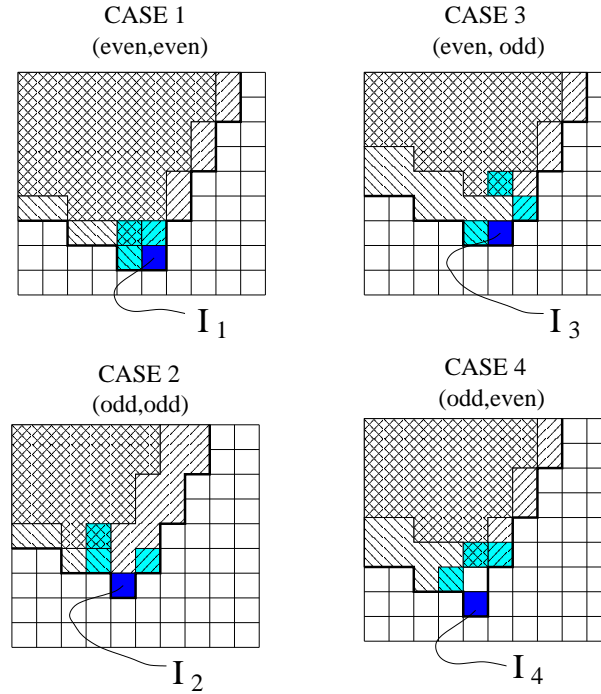
In our implementation we used a *unit* Integral Image that counts the number of pixels in each area that composes a Haar-like feature. Although this requires extra look-ups it allows to rapidly compute the correction factors. One advantage of this approach is that the unit Integral Image does not change when new frames are being assessed, saving precious time. The *unit* Integral Image is only necessary for the the rotated features ( $45^\circ$ ,  $26.5^\circ$  and  $63.5^\circ$ ) mainly because for the normal ( $0^\circ$ ) case the number of pixels can be based on the width and height of the feature.

The second error source cannot be compensated without a more complicated approach such as computing sub-pixel values for the areas. This is not usually a good approach because it loses the advantage of computing areas very rapidly with the assistance of the Integral Images.

The third error occurs when comparing the same image with its counterpart after a processing operation such as scaling and rotation. Anti-aliasing techniques applied when scaling images can cause variations even if the features can fit the position and the sizes perfectly. Although this error is not directly related to the features' definition or implementation itself, it is an important source of errors during the detection phase or testing.

## 2.2 Computing Haar-like features at $45^\circ$ and $26.5^\circ$

Twisted (or "tilted") Haar-like features were proposed by Lienhart et al. (9) to test the hypotheses that a stronger classifier can be built if  $45^\circ$  features were included in the set. The normal features

Figure 3: Computing the Integral Image for  $26.56^\circ$  recursively

and the twisted features are not mathematically equivalent in digital processing due to the fact that the twisted Integral Image needs slightly distorted rectangles to correctly compute an area. Figure 2 shows an example where two features, one normal and the other twisted, would compute similar areas. Notice the double pixel on two vertices of the twisted feature. This is necessary in order to get the correct alignment so the Integral Image can supply the correct sum of pixels (9). Yet another approximation has to be made when computing the width and the height of the twisted feature, as this calculation often yields a sub-pixel value.

Unfortunately for other angles the solution is not trivial, as the shapes of the areas covered by one element of the Integral Image changes depending on the parity of the coordinates. However, there is a partial solution for the case where the ratio is 1:2 or 2:1. As these angles also allow the computation on the other quadrants, four Integral Images will suffice to divide  $360^\circ$  in 16 regions (only approximately symmetric).

Next we formalise the calculation of the Integral Image for the case of  $26.5^\circ$ . The other case ( $63.5^\circ$ ) is analogous. Each element of the Integral Image will cover different areas of the image. There are four cases that depend on the parity of the coordinates ( (even,even), (even,odd), (odd,even) and (odd,odd) coordinates). Figure 3 shows the four cases and the areas that they cover respectively. The Integral Image can be computed recursively using the following equations:

$$I_1(x,y) = I_{(x-1,y)} + I_{(x,y-1)} - I_{(x-1,y-1)} + im_{(x,y)} \quad (2.1)$$

$$I_2(x,y) = I_{(x+1,y-1)} + I_{(x-1,y-1)} - I_{(x-1,y-2)} + im_{(x,y)} \quad (2.2)$$

$$I_3(x,y) = I_{(x-1,y)} + I_{(x+1,y-1)} - I_{(x,y-2)} + im_{(x,y)} \quad (2.3)$$

$$I_4(x,y) = I_{(x-1,y-1)} + I_{(x+1,y-2)} - I_{(x,y-2)} + im_{(x,y)} + im_{(x,y-1)} \quad (2.4)$$

Where:  $I_1$  has an (even,even) coordinate,  $I_2$  has (odd,odd),  $I_3$  has (even,odd) and  $I_4$  has (odd,even). The pixels of the original image  $im_{(x,y)}$  might be reallocated one position to avoid negative indexes.

The computation of the Haar-like features at  $26.5^\circ$  also demands that the alignment of the 4 points are coherent with the creation of the Integral Image. Extra operations to analyse the parity of the coordinates are necessary. The possible combination of the parities of 4 points mount to 64 cases. Most cases do not need any correction. About a dozen cases need a displacement of one position on one or two points so the alignment is respected.

### 2.3 Converting features to generic angles

We propose using a function of the values of two features to approximate the value for a feature rotated at a generic angle. We call this approach *pair of equivalent features* (PEF). This is achieved using a weighted sum of the two equivalent features and a conversion of feature positions, feature sizes and feature types. Figure 4 shows how this conversion is done for the case where normal and  $45^\circ$  features are used. For an angle  $\alpha$  the new features has to be positioned on a new kernel, larger in size to accommodate the rotation of the second feature. For example, for angles between  $0^\circ$  and  $45^\circ$  formula 2.5 applies. For angles larger than  $45^\circ$  there is also a feature type change to be made. For example, the PEF of a type 0 feature for angles between  $0^\circ$  and  $45^\circ$  will be a normal feature of type 0 and a twisted feature of type 10. If the angle is between  $45^\circ$  and  $90^\circ$ , the PEF will be a normal feature of type 1 and a twisted feature of type 10 (figure 5). For other angles the PEF may be computed from other pairs, in some cases involving a change of sign.

The value of the PEF can be approximate by the following equation:

$$V = V_{normal} \cdot (45^\circ - \alpha) / 45^\circ + V_{twisted} \cdot \alpha / 45^\circ \quad (2.5)$$

Where:  $V_{normal}$  is the Value for the normal feature,  $V_{twisted}$  is the value for the twisted feature and  $V$  is the weighted average that depends on the angle  $\alpha$  (between  $0^\circ$  and  $45^\circ$ ).

Analogous to the case where the PEF is computed with a normal and a  $45^\circ$  angle, one can approximate the feature value for any angle between  $0^\circ$  and  $26.5^\circ$ :

$$V = V_{normal} \cdot (26.5^\circ - \alpha) / 26.5^\circ + V_{26} \cdot \alpha / 26.5^\circ \quad (2.6)$$

Where:  $V_{normal}$  is the Value for the normal feature,  $V_{26}$  is the value for the feature at  $26.5^\circ$  and  $V$  is the weighted average that depends on the angle  $\alpha$  (between  $0^\circ$  and  $26.5^\circ$ ).

### 2.4 PEF errors

A Haar-like feature value will depend on the distribution of the pixels in the area where it is applied. The maximum value for many types of Haar-like features occurs when an edge is located in the middle of the feature. In order to maximise the absolute value of the feature it is also necessary that all the values of the pixels on one side of the edge are zeros and all the other pixel values are maximum (for example, 255 in a grey scale image).

From this point lets call the maximum value of a feature at a generic scale to be MAX, so that any feature value can vary from -MAX to MAX. Lets suppose that we could compute a *type 0* Haar-like feature at an angle of  $22.5^\circ$  over an edge presented by the image. We assume for this example that the ratio width/height is approximately 2:5. The theoretical value for the figure 6-a is MAX. The PEF value is  $\text{abs}(\text{MAX}/2)$  (figure 6-b), as both the normal feature and the twisted feature will yield the same value MAX/2. For figure 7-a the value of the feature would be zero. For the normal features is MAX/8 and for the twisted features it is -MAX/8 (figure 7-b). Therefore the PEF value is zero, which is the required result. In the extreme cases an error of about 50% in relation to MAX is expected. Variations on these errors might be expected for different width/height ratios.

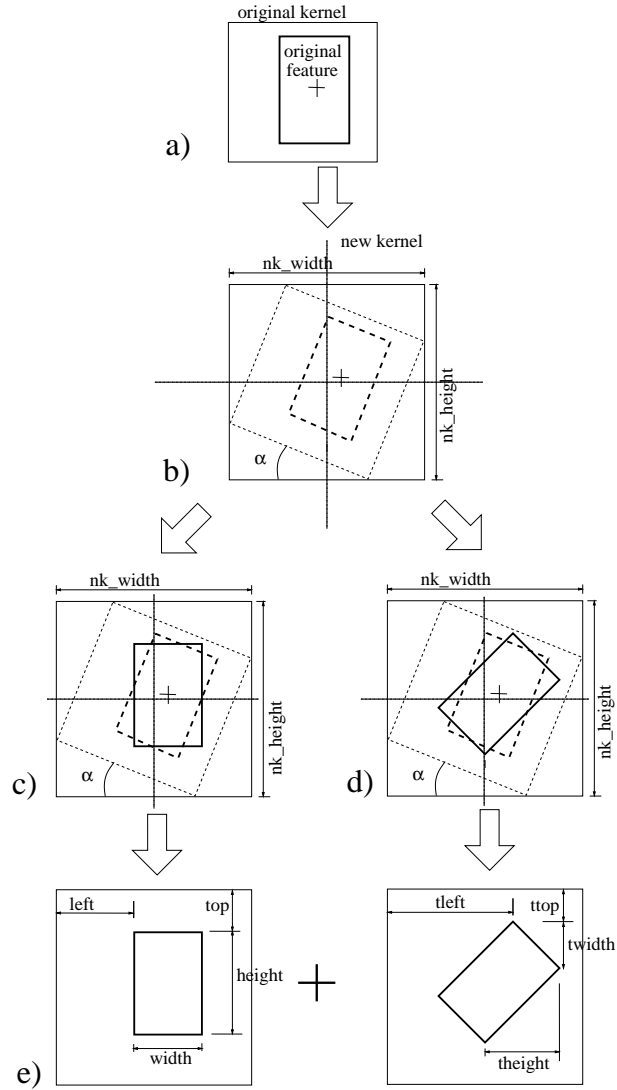


Figure 4: a) The original feature positioned in the original kernel. b) The new kernel size and the rotation angle are shown. c) The position for the new normal feature is computed. d) The size and the position is computed for the twisted feature. e) The resulting Pair of Equivalent Features (PEF).

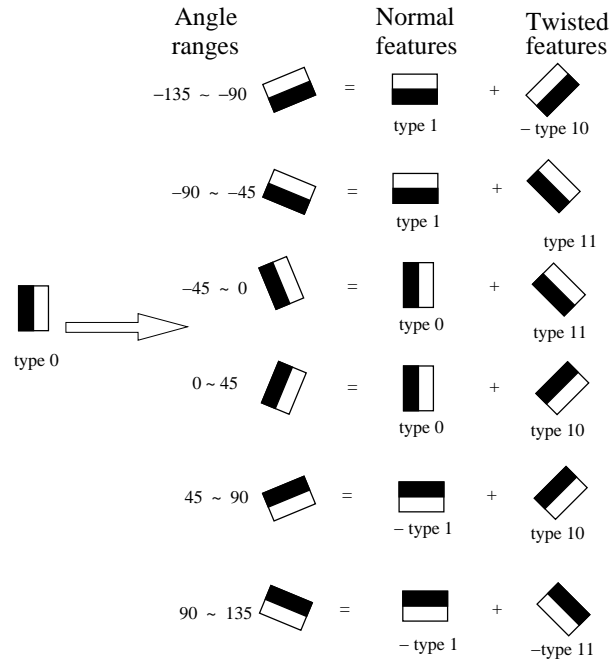


Figure 5: An example of PEFs for a type 0 feature considering a number of angle ranges.

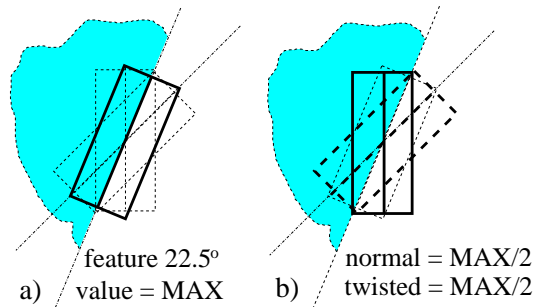


Figure 6: Case where the value of the 22.5° feature should be MAX.

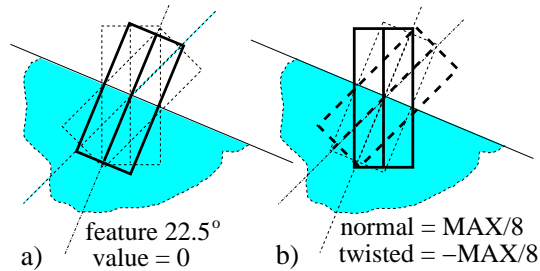


Figure 7: Case where the value of the 22.5° feature should be zero.

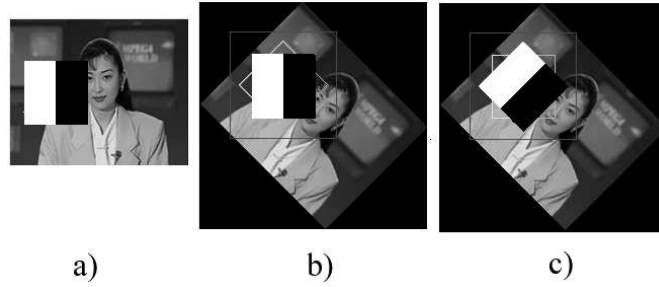


Figure 8: Measuring errors for PEFs. a) the original normal feature b) the converted normal feature c) the converted  $45^\circ$  feature.



Figure 9: First frame of Akiyo sequence and the chessboard images.

### 3 Experimental Results and Discussion

#### 3.1 Experiment 1: error analysis for PEFs using $0^\circ$ and $45^\circ$

In order to assess the impact of the approximation, an experiment was carried out using several natural images as well as binary images where the edges would call for the maximum value of the features. The images were rotated to angle  $\alpha$ . Features at all possible scales and positions were computed in the original image. The PEFs for each of those features were also computed. Figure 8 shows the error measurement approach for the first frame of the Akiyo sequence at  $45^\circ$ . For each angle of rotation several thousand PEFs were computed this way.

The errors were calculated as a percentage of the maximum possible value ( $MAX$ ) for that feature type at that scale ( $F$  is the original feature value and  $V$  is the PEF value for that angle):

$$error = \frac{|F - V|}{MAX} \quad (3.1)$$

We present here the results for two images, the first frame of the Akiyo sequence and a Chessboard (figure 9). The first image is significant because it contains a face, a common object used in detection algorithms. The second image yields large errors because it contains well defined vertical and horizontal edges. A small deviation on the position of the converted features may cause relatively large errors.

The feature size was  $20 \times 20$  pixels on a kernel of the same size. The kernel size for the PEF was  $34 \times 34$  pixels. Figures 10, 11, 12 and 13 show the maximum errors for both images using different type of Haar-like features. As expected the largest errors were around the region of  $22.5^\circ$ . Looking at the errors at  $45^\circ$  notice that they decrease almost to the same levels of those at  $1^\circ$ . This indicates that the twisted features can be used to convert normal features to  $45^\circ$  with good accuracy. In fact we tested classifiers trained with normal features, converted it to twisted features and successfully used it to detect faces at  $45^\circ$ .

The maximum error for Akiyo was 17% for type 6 feature (figure 10). For the chessboard



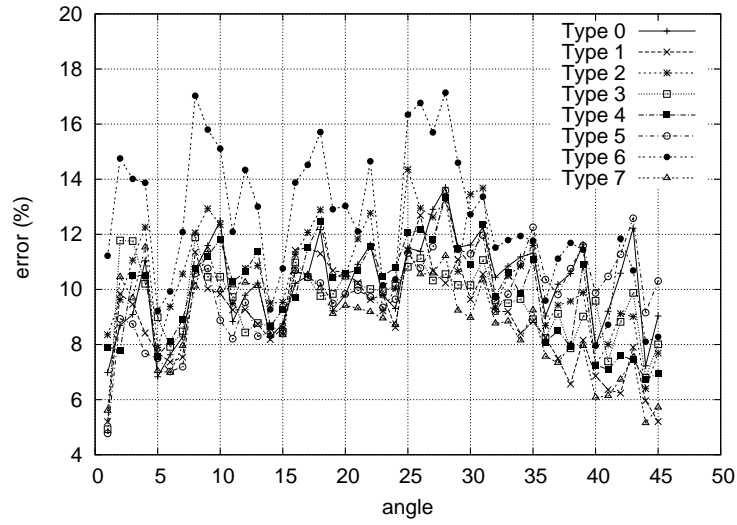


Figure 10: Maximum error vs angle for Akiyo (normal+45° PEFs).

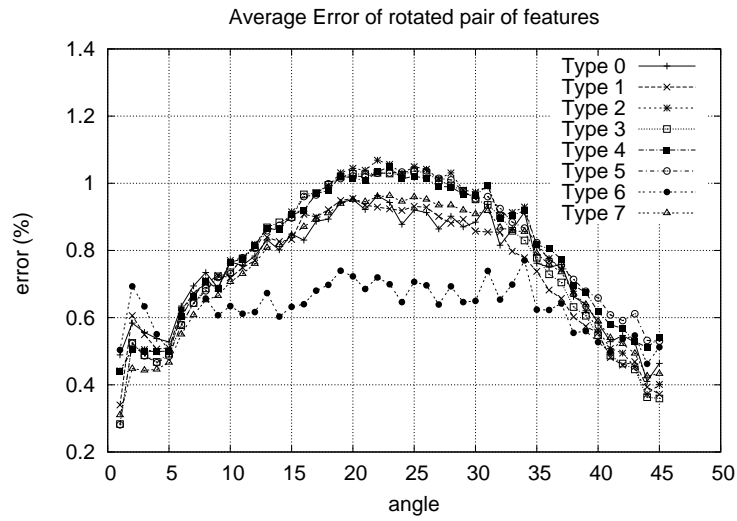


Figure 11: Average error vs angle for Akiyo (normal+45° PEFs).

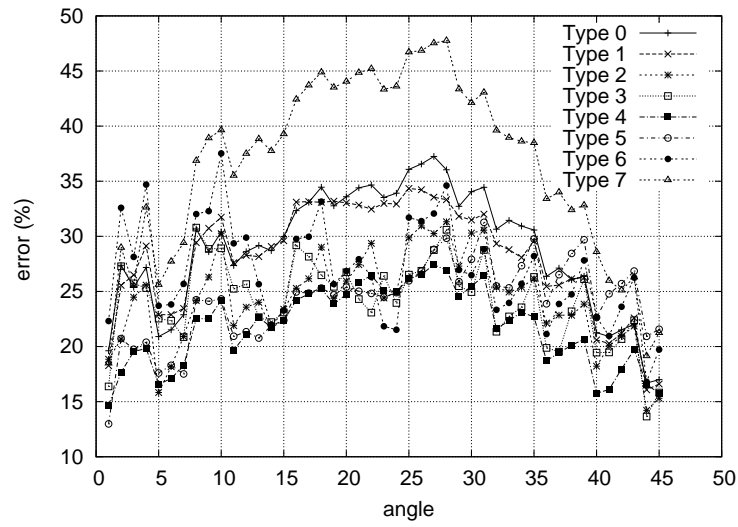


Figure 12: Maximum error vs angle for Chessboard (normal+45° PEFs).

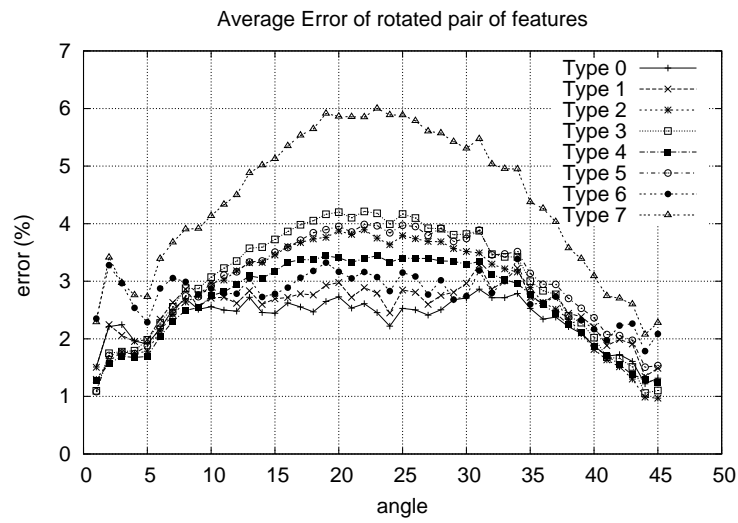


Figure 13: Average error vs angle for Chessboard (normal+45° PEFs).

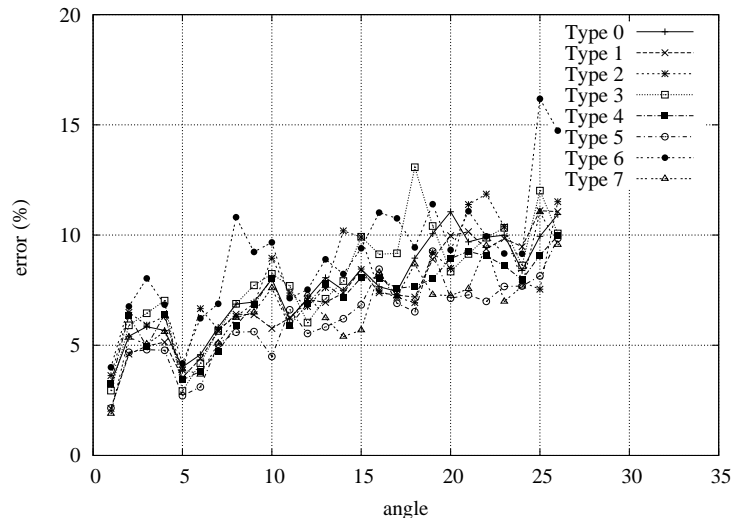


Figure 14: Maximum error vs angle for Akiyo (normal+26.5° PEFs).

image the maximum error was 47% for type 7 features (figure 12). The maximum error in this case almost reached the theoretical value of 50%. The average errors, based on the absolute value of the PEFs, indicate that for most features the PEF values are actually very close to the original feature. For Akiyo, the maximum average error was around 1.1% for feature type 2 (figure 11). For the chessboard the maximum average error was 6% for type 7 features (figure 13).

### 3.2 Experiment 2: error analysis for PEFs using 0° and 26.5°

In this experiment the third group of features on figure 1 was implemented. Figures 14 and 15 show the maximum errors obtained for Akiyo and for Chessboard. Figures 16 and 17 show the average errors. For some angles the errors improved when comparing to the same angles in experiment 1.

The maximum error for Akiyo was 12% for type 3 feature (figure 14). For the chessboard image the maximum error was 34% for type 3 features (figure 15). The maximum errors were close to those obtained in experiment 1. For Akiyo, the maximum average error was around 1.4% for feature type 1 (figure 16). For the chessboard the maximum average error was 6% for type 7 features (figure 17).

However it is clear that the errors at angles around 26° should be much smaller. The origin of these errors are related to the position calculations during the feature conversion. The positioning of the equivalent features at these angles are much more sensitive than the ones in experiment 1. Any small deviation on one of the points in the Integral Image result in larger errors. The solution for this problem consists in improving the rounding algorithm to consider the best point depending on the circumstances of the area being computed (position, scaling, parity of the coordinates).

## 4 Conclusions

In this paper a new approach for a rotational invariant Viola-Jones detector was developed. The proposed method allows to convert a previously trained classifier to work at any angle, so rotated objects can be detected without specifically training the classifier for that angle. The accuracy of the features computed using the method was tested using different images through two experiments.

The first experiment showed that twisted features can successfully convert a normal feature to angles close to 45°. However for angles at the vicinity of 22.5° the PEFs suffer from large errors

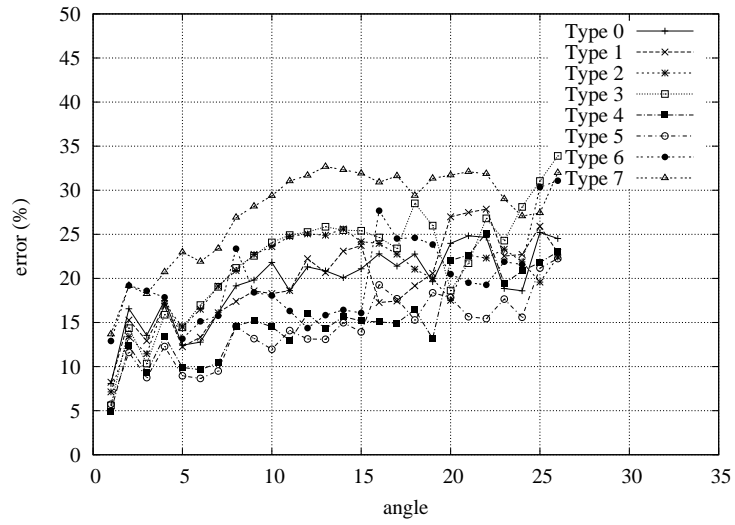


Figure 15: Maximum error vs angle for Chessboard (normal+26.5° PEFs).

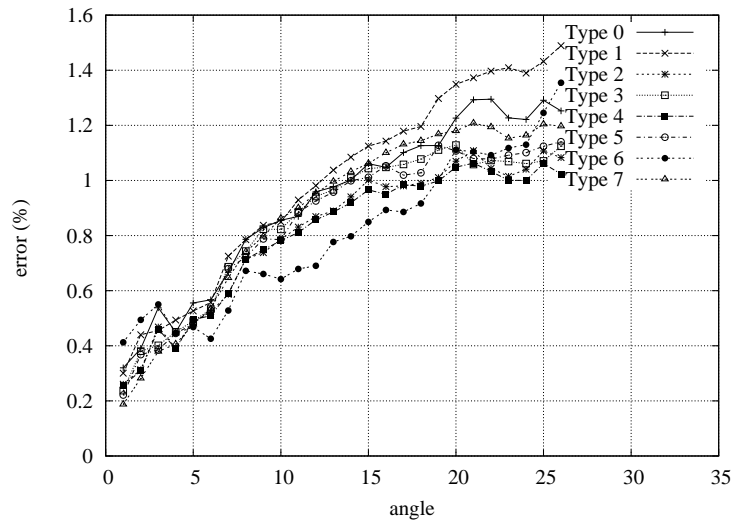


Figure 16: Average error vs angle for Akiyo (normal+26.5° PEFs).

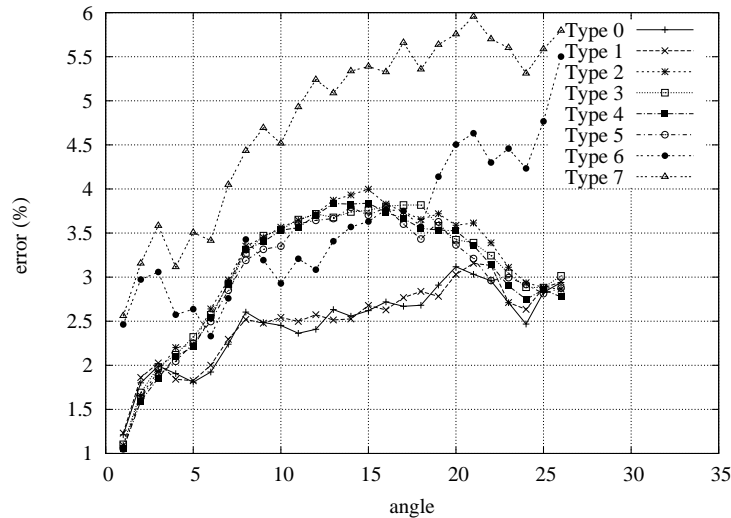


Figure 17: Average error vs angle for Chessboard (normal+26.5° PEFs).

that might affect the accuracy of the classifier. The second experiment showed that although some of the errors are smaller than in experiment 1, the implementation needs some improvement to successfully compute equivalent features at angles of 26.5°.

The limitations for this approach occur when very long Haar-like features are used to compose a classifier, in which case the errors associated with the PEF are too large to be considered. However it is possible to train classifiers using a more limited set of Haar-like features to avoid this setback.

For future work we intend to implement an error correction approach regarding the positioning and sizes of the features that compose a PEF. We are also going to assess and analyse the hit ratios of actual classifiers using the PEF approach.

## References

- [1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *CVPR01*, pp. I:511–518, IEEE, December 2001.
- [2] M. J. Jones and P. Viola, “Robust real-time object detection,” Tech. Rep. CRL-2001-1, Hewlett Packard Laboratories, February 2001.
- [3] P. Viola and M. Jones, “Robust Real-Time face detection,” in *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01)*, (Los Alamitos, CA), pp. 747–747, IEEE Computer Society, July 2001.
- [4] G. Guo and H. Zhang, “Boosting for fast face recognition,” Tech. Rep. MSR-TR-2001-16, Microsoft Research (MSR), Feb. 2001.
- [5] M. Kolsch and M. Turk, “Analysis of rotational robustness of hand detection with a viola-jones detector,” in *ICPR04*, pp. III: 107–110, 2004.
- [6] S. Mitri, K. Pervlz, H. Surmann, and A. Nchter, “Fast color-independent ball detection for mobile robots,” *Mechatronics and Robotics*, pp. 900–905, September 2004.
- [7] T. Burghardt, B. Thomas, P. J. Barham, and J. Calic, “Automated visual recognition of individual african penguins,” in *Fifth International Penguin Conference*, (Ushuaia, Tierra del Fuego, Argentina), September 2004.

- [8] M. J. Jones and P. Viola, “Fast multi-view face detection,” Tech. Rep. TR2003-96, MERL, July 2003.
- [9] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *ICIP02*, pp. I: 900–903, September 2002.
- [10] Y. Freund and R. E. Schapire, “A short introduction to boosting,” *Journal of Jap. Society for Art. Intell.*, vol. 14, no. 5, pp. 771–780, 1999.
- [11] R. Lienhart, A. Kuranov, and V. Pisarevsky, “Empirical analysis of detection cascades of boosted classifiers for rapid object detection,” in *DAGM03*, (Magdeburg, Germany), pp. 297–304, September 2003.
- [12] S. Z. Li and Z. Zhang, “Floatboost learning and statistical face detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1112–1123, September 2004.
- [13] B. McCane and K. Novins, “On training cascade face detectors,” in *Image and Vision Computing New Zealand*, (Palmerston North), pp. 239–244, 2003.