1-1-2009

Face detection using generalised integral image features

Alister Cordiner
Digisensory Technologies Sydney, alc456@uow.edu.au

Philip Ogunbona
University of Wollongong, philipo@uow.edu.au

Wanqing Li
University of Wollongong, wanqing@uow.edu.au

Face detection using generalised integral image features

Abstract

This paper proposes generalised integral image features (GIIFs) for face detection. GIIFs provide a richer and more flexible set of features than Haar-like features. Due to the large set of possible GIIFs, a genetic algorithm is developed to select the feature space for the optimal weak classifiers. Experimental results have shown that this method is able to improve face detection accuracy.

Keywords

face, generalised, detection, integral, features, image

Disciplines

Physical Sciences and Mathematics

Publication Details

Cordiner, A., Ogunbona, P. & Li, W. (2009). Face detection using generalised integral image features. International Conference on Image Processing (pp. 1229-1232). Cairo, Eygpt: IEEE.

FACE DETECTION USING GENERALISED INTEGRAL IMAGE FEATURES

Alister Cordiner[1,2]

[1]Digisensory Technologies
Sydney, Australia

Philip Ogunbona[2], *Wanqing Li*[2]

[2]Advanced Multimedia Research Lab
ICT Research Institute, University of Wollongong

ABSTRACT

This paper proposes *generalised integral image features* (GIIFs) for face detection. GIIFs provide a richer and more flexible set of features than Haar-like features. Due to the large set of possible GIIFs, a genetic algorithm is developed to select the feature space for the optimal weak classifiers. Experimental results have shown that this method is able to improve face detection accuracy.

Index Terms— integral image, GIIF, face detection

1. INTRODUCTION

Haar-like features have been widely used in object detection, especially face detection [1]. In essence, they encode spatial intensity differences across an image. The value of a single Haar-like feature can be expressed as:

$$z_n = \mathbf{x}_n^T \mathbf{h} \tag{1}$$

where z_n is the feature value, $\mathbf{x}_n$ is the input image and $\mathbf{h}$ is the Haar-like feature at a single location and scale padded with zeros to match the size of $\mathbf{x}_n$. Viola and Jones suggested the use of *integral images* to act as a form of look-up table to speed up the calculation of z_n. The integral image transform is a recursive linear filter that finds the 2-D discrete antiderivative, defined as:

$$\begin{aligned} y[u,v] &= \sum_{u' \leq u,\, v' \leq v} x(u',v') \\ &= x[u,v] + y[u,v-1] + y[u-1,v] \\ &\quad -y[u-1,v-1] \end{aligned} \tag{2}$$

where x corresponds to the input image and y is the integral image. Each pixel in this image is the sum of all of the pixels to the left of and above the corresponding pixel in the image.

Once the integral image is calculated, Equation (1) can be replaced with:

$$z_n = \mathbf{y}_n^T \mathbf{g} \tag{3}$$

where $\mathbf{y}_n$ is the integral image representation and, in this paper, $\mathbf{g}$ is referred to as the *look-up vector*. The vector $\mathbf{g}$ can

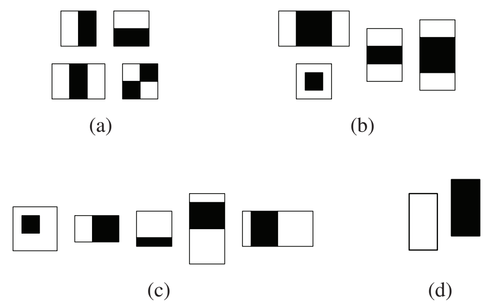


Fig. 1. Some examples of Haar-like features used to represent $\mathbf{h}$. (a) Original feature set [1]. (b) Extended (non-rotated) feature set [2]. (c) Asymmetric feature set [3]. (d) Dissociated dipoles [4].

be calculated by convolving the Haar-like feature with a 2-D derivative filter kernel to find the discrete derivative:

$$g(u,v) = h(u,v) * \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix} \tag{4}$$

The sum of any rectangular region can then be found with just four look-ups in $\mathbf{y}$.

This paper proposes the generalised integral image features (GIIFs) by allowing $\mathbf{g}$ to take any values. The Haar-like features are thus a subset of the GIIFs. Due to the large feature set provided by the GIIFs, a genetic algorithm is developed to select the features for the optimal weak classifiers. Application of the GIIFs to face detection has shown that GIIFs can improve detection accuracy.

The rest of this paper is laid out as follows: section 2 gives a brief overview of related work. The proposed GIIFs and generic algorithm (GA) to select the feature space are described in section 3. Section 4 describes the experimental results, and section 5 concludes the paper and outlines possible future extension.

2. RELATED WORK

One of the most successful applications of Haar-like features to face detection was described by Viola and Jones [1], using the set of features shown in Figure 1(a). Many other authors have explored using alternate sets of Haar-like features to achieve improved accuracy. An extended feature set was later proposed by Lienhart and Maydt [2] to improve accuracy and it included a set of rotated features. The non-rotated features from this set are shown in Figure 1(b). Asymmetric features were proposed by Ramirez and Fuentes [3] and are shown in Figure 1(c). These remove the constraint of symmetry on $\mathbf{h}$ and are claimed to significantly improve detection accuracy. Dissociated dipoles are features that are disjointed [4], shown in Figure 1(d). They contain two rectangular regions, one positive and one negative, but unlike the traditional Haar-like features there is no constraint that these regions must be adjacent.

Many of these Haar-like feature variants generate very large feature spaces; the asymmetric Haar-like features in Figure 1(c) produce over 2×10^8 features for a single 24×24 image [3], and the dissociated dipoles in Figure 1(d) generate 2^{28} features [5]. The exhaustive feature selection approach used by Viola and Jones to generate the weak classifiers is impractical in such high dimensional feature spaces. For this reason, more efficient methods of feature selection have been proposed, many of them based on evolutionary learning.

For the asymmetric features in Figure 1(c), the authors parametrised the features into a vector composed of their width and height and used a genetic algorithm to select the optimal features [3]. Baró and Vitrià [5] proposed probabilistic evolutionary learning to search for optimal dissociated dipoles, parametrising the features into a vector composed of position, scale and type. Two additional Haar-like features to Viola and Jones including one weighted dissociated dipole with varying feature weights $\mathbf{h} \in [-4, +4]$ were proposed by Treptow and Zell [6]. Evolutionary search is used to select the optimal weak classifier at each stage using vectors encoding feature type and position. Abramson and Steux [7] proposed YEF (Yet Even Faster) features which compare sets of individual pixels rather than rectangular regions. They used a genetic algorithm at each round of boosting to select the optimal feature.

However, despite the more diverse Haar-like feature sets many authors have proposed, and the alternatives to exhaustive search that make such feature sets practical, most of these techniques still require a specific Haar-like feature set to be hand selected. The YEF features do not require this heuristic approach, however there is no obvious way to apply this detector at multiple resolutions without a multiresolution pyramid (they suggest training multiple detectors at different scales) and their detector, while twice as fast as Viola and Jones, is not as accurate.

3. GIIFS AND FACE DETECTION

3.1. GIIFs

In Figure 4, an example of a single Haar-like feature $\mathbf{h}$ and the corresponding look-ups in the integral image required to calculate this feature $\mathbf{g}$ are shown. Each feature will have a different corresponding look-up vector $\mathbf{g}$ for each location and size. However, the fact that a specific set of Haar-like features is defined *a priori* means that we do not consider every possible look-up vector $\mathbf{g}$; instead, we only consider the subset of all possible vectors based on the set of Haar-like features.

The idea of the GIIFs is that $\mathbf{g}$ should not be constrained based on a set of heuristically or empirically selected Haar-like features. Instead, $\mathbf{g}$ is allowed to take on potentially any value. If there are n pixels in an image, and we allow each element of the look-up vector $\mathbf{g}$ to take a total of λ values, then there are λ^n possible features. For example, for a 24×24 sub-window size with $\mathbf{g} \in \{-1, 0, +1\}$ (i.e. $\lambda = 3$ possible values for each element of the vector $\mathbf{g}$), this would result in 6×10^{274} possibilities. The solution proposed here is to restrict ourselves to sparse vectors $\mathbf{g}$. This means that the look-up vector $\mathbf{g}$ should contain mostly zero entries, equating to a small number of look-ups required to calculate a feature value. If we allow $\mathbf{g}$ to contain less than or equal to k non-zero entries and there are n pixels per image, the total number of possibilities is:

$$p = \sum_{k'=1}^{k} \left[(\lambda - 1)^{k'} \times \binom{n}{k'} \right] = \sum_{k'=1}^{k} \frac{n!(\lambda - 1)^{k'}}{k'!(n - k')!} \qquad (5)$$

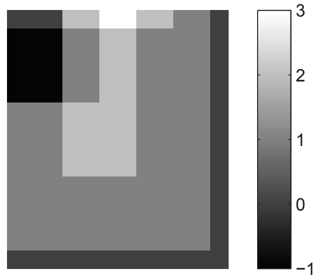
For example, if we only allow a single non-zero entry in the look-up table and use the constraint $\mathbf{g} \in \{-1, 0, +1\}$ (i.e. $\lambda = 3$), then this means that $p = 1,152$ possible feature values, which becomes a feasible feature space to search. Similarly, if we we allow 6 or less non-zero entries as in Figure 4(a), there are 24×24 pixels in the image and $\mathbf{g} \in \{-1, 0, +1\}$, this results in $p \approx 3.2 \times 10^{15}$ possible combinations. This feature space is still significantly larger than that generated by the Haar-like features of Viola and Jones, which require from 6 to 9 non-zero entries with values $\mathbf{g} \in \{-2, -1, 0, +1, +2\}$) and provide a pool of only 1.8×10^5 features to select from at each round of training [1], but is a tractable problem for a genetic algorithm to solve. In addition, many of the Haar-like features from Figure 1 are just a subset of the GIIFs.

3.1.1. Translation invariant GIIFs

So far the discussion of GIIFs has assumed that a single detector is applied at a single location and size. Viola and Jones use a scanning sub-window to detect objects at different locations and sizes in the input image. There are two possible approaches to using integral images with sub-windows: (i) calculate the integral image once on the entire input image, then slide the detector sub-window over the integral image; or (ii) calculate a new integral image separately for each sub-window as it sweeps the input images. The first method is far more efficient, however the integral image pixel values are

0	0	0	0	1	0	0
0	0	0	0	0	0	0
0	−1	−1	0	0	0	0
0	0	0	0	0	0	0
0	−1	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0

(a) Look-up vector $\mathbf{g}$



(b) Feature $\mathbf{h}$

Fig. 2. Example integral image look-ups used to calculate a single generalised integral image feature value.

0	0	0	0
0	0	1	0
0	2	0	0
0	0	0	0

3	−2	−1	0
−1	0	1	0
−2	2	0	0
0	0	0	0

(a)

0	0	0	0
0	−1	0	0
0	0	1	0
0	0	0	0

0	1	−1	0
1	−1	0	0
−1	0	1	0
0	0	0	0

(b)

Fig. 3. Two example conversions from a GIIF g (left) to a translation invariant GIIF $\tilde{g}$ (right)

a function of their location within the image, and so are not translation invariant.

If all of a look-up vector's columns and rows sum to zero, the values of the feature will cancel out the regions above and to the left of the sub-window in the integral image, and therefore the feature is translation invariant. Viola and Jones define Haar-like features that are composed of rectangular regions that always sum to zero and so are inherently translation invariant. Any arbitrary GIIF can be made translation invariant by modifying the values in the first row and first column of the 2-D look-up vector $g(u, v)$ such that its columns and rows sum to zero, using the following piece-wise equation:

$$\tilde{g}(u,v) = \begin{cases} \sum_{u'=2}^{w} \sum_{v'=2}^{h} g(u',v') & \text{if } u = 1 \text{ and } v = 1 \\ -\sum_{u'=2}^{w} g(u',v) & \text{if } u = 1 \text{ and } v > 1 \\ -\sum_{v'=2}^{h} g(u,v') & \text{if } u > 1 \text{ and } v = 1 \\ g(u,v) & \text{otherwise} \end{cases} \tag{6}$$

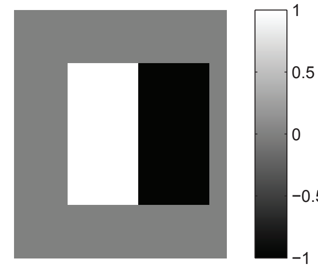
where $\tilde{g}$ is the translation invariant GIIF and w and h are the width and height of the feature g. This step is applied at three stages during the GA search: (i) after the creation of the initial random population; (ii) after the mutation function is applied; and (iii) after the crossover function is applied. Some examples of translation invariant GIIFs are shown in Figure 3.

3.2. Training the Face Detector with GA-based Search

Training is used to statistically select appropriate features that are suitable to represent the object being detected. Rather than performing training on the Haar-like feature outputs and utilising the integral image representation merely as a computational short-cut, the training is performed directly on the integral image itself.

0	0	0	0	0	0	0
0	1	0	−2	0	1	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	−1	0	2	0	−1	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(a) Look-up vector $\mathbf{g}$



(b) Feature $\mathbf{h}$

Fig. 4. Example integral image look-ups used to calculate a single Haar-like feature value.

AdaBoost is used to generate an ensemble of weak classifiers, each trained on individual features. Since GIIFs allow a much richer set of representations than Haar-like features and also result in a much larger feature space to be searched during training, the exhaustive search as adopted by Viola and Jones is no longer feasible. In this paper, we propose a GA-based search algorithm. The pseudocode for the GA procedure for generating a weak classifier is shown in Algorithm (1). Viola and Jones select the look-up vector $\mathbf{g}$ at each round that minimises the weighted error rate. In the proposed method, a two-part objective function J_m is defined to minimise the weighted training error and maximise the sparsity (number of zero elements) of $\mathbf{g}$ and allows the relative weighting between the error and sparsity $\alpha \in [0, 1]$ to be varied:

$$\begin{aligned} J_m(\mathbf{g}) \quad &= \quad \alpha \left[\frac{1}{N} \sum_{n=1}^{N} w_n \left(f_m(z_n) - c_n \right)^2 \right] + \\ & \qquad (1-\alpha) \left[\frac{1}{k} \sum_{i=1}^{k} S[g(i)] \right] \end{aligned} \tag{7}$$

where f_m, $c_n \in \{-1, 1\}$ are the weak classifier output and class label and $S(x)$ is a function that outputs zero if x equals zero and a unit value if it is non-zero. Therefore, rather than explicitly restricting the classifier to a pre-defined number of look-ups in order to reduce the search space, the training favours sparse look-up vectors as it searches the feature space because it is incorporated into the objective function.

Note that the translation invariant GIIF $\tilde{g}$ described in section 3.1.1 is usually denser than the original GIIF g. Because the fitness function in Equation (7) incorporates sparsity, the GA search will be less effective at reducing the error rate. Nevertheless, the translation invariant GIIF is far more efficient and better suited to applications where real-time detection is required.

Algorithm 1 Training a single weak classifier using GA

1. Randomly generate initial population
2. Repeat for n generations:
   (a) Train a weak classifier on every chromosome in the population
   (b) Evaluate the fitness of each classifier using Equation (7)
   (c) Select the best subset of classifiers/chromosomes
   (d) Mutate and crossover this subset to generate the next population of chromosomes
3. Output the best weak classifier

3.3. Computational complexity

The computational complexity of classification using the GA-based GIIF approach is equivalent to the approach of Viola and Jones. However, unlike the exhaustive feature selection method of Viola and Jones, evolutionary searches can have a variable training time. If we set a maximum generations with a population size of b at each generation, then a maximum of ab features must be evaluated at each round. In practice, fewer may be evaluated since the GA may exit early if a stopping criterion is met.

4. EXPERIMENTAL RESULTS

The performance of the GA-based GIIF face detector was compared to the Viola and Jones face detector using the Haar-like features from Figure 1(a) on the Bio ID face database (with randomly generated non-face samples cropped from natural images) using the following parameters: population size of 2500, maximum of 2500 generations, mutation rate of 0.1, crossover rate of 0.9 and look-up vector range $\mathbf{g} \in [-1, +1]$. The GIIF detector was tested with and without the translation invariance step incorporated during the training. The resulting ROC curves are shown in Figure 5. The GIIF method without translation invariance performed the best, with an Area Under Curve of 0.997, followed by the GIIF method with translation invariance at 0.990 and the Viola and Jones method at 0.987. The equal error rates were 2.7%, 5.0% and 6.3% respectively.

5. CONCLUSION AND FUTURE EXTENSIONS

GIIFs allow a much richer set of representations than Haar-like features and experimental results indicate that they are able to achieve higher face detection accuracy. The translation invariant GIIFs achieve lower accuracy, but are far more efficient if a scanning sub-window is used.

This technique may find other applications in the wider area of object recognition. Although Haar-like features are suitable for face detection, the GIIF approach may be able to generalise better to diverse object detection tasks by learning appropriate feature sets.

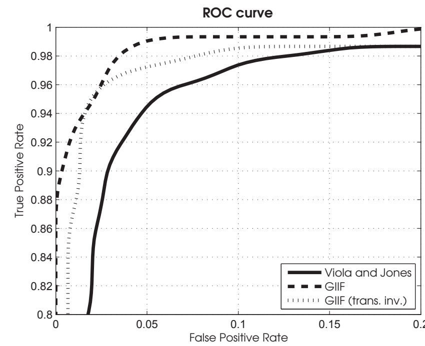


Fig. 5. Comparison of the ROC curves

An interesting extension of the work of Viola and Jones was provided by Lienhart and Maydt [2]. They introduced the idea of rotated features, which are calculated using rotated integral images. Future work could involve generating GIIFs from rotated integral images.

6. REFERENCES

[1] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*. 2001, vol. 1, pp. 511–518, IEEE Computer Society.

[2] Rainer Lienhart and Jochen Maydt, "An extended set of Haar-like features for rapid object detection," in *IEEE International Conference on Image Processing*, 2002, vol. 1, pp. 900–903.

[3] Geovany A. Ramirez and Olac Fuentes, "Multi-pose face detection with asymmetric Haar features," in *IEEE Workshop on Applications of Computer Vision*, Copper Mountain, CO, USA, January 2008, pp. 1–6.

[4] Benjamin J Balas and Pawan Sinha, "STICKS: Image-representation via non-local comparisons," *Journal of Vision*, vol. 3, no. 9, pp. 12, 2003.

[5] Xavier Baró and Jordi Vitrià, "Evolutionary object detection by means of naïve bayes models estimation," in *Applications of Evolutionary Computing*, 2008, number LNCS 4974, pp. 235–244.

[6] André Treptow and Andreas Zell, "Combining Adaboost learning and evolutionary search to select features for real-time object detection," in *Congress on Evolutionary Computation*, June 2004, vol. 2, pp. 2107–2113.

[7] Yotam Abramson and Bruno Steux, "YEF real-time object detection," in *International Workshop on Automatic Learning and Real-Time*, Siegen, Germany, September 2005, pp. 5–13, University of Siegen.