Abstract. We demonstrate a technique allowing for constant-time calculation of low order Fourier moments, applicable in detection tasks. Real and imaginary parts of the moments can be used as features for machine learning and classification of image windows. The technique is based on a set of special integral images, prepared prior to the scanning procedure. The integral images are constructed as cumulative inner products between the input image and suitable trigonometric terms. Additional time invested in the preparation of such integral images is amortized later at the stage of scanning. Then, the extraction of each moment requires only 21 operations, regardless of the number of pixels in the detection window, and thereby is an $O(1)$ calculation.

As an application example, face detection experiments are carried out with detectors based on Haar-like features serving as opponents to the proposed Fourier-based detectors.

1 Introduction

Constant-time computational complexity is the most attractive complexity for a computer scientist. Unfortunately, favourable opportunities to apply algorithms of that complexity are rare — typically, they pertain to some selected data structures e.g. hash tables, Union-Find\footnote{For strictness: the ‘Find’ operation in this data structure is of amortized complexity $O(\log^* n)$ — iterated logarithm of $n$. Wherein $\log^* n$ is not greater than 5 for all quantities $n$ observable in the universe; in particular, $\log^* 2^{65536} = 5$.} \cite{2} and constitute a narrow fragment of a larger software. Often, one deals in fact with a so-called amortized constant-time complexity. This means that in the company of essential operations, performed are also some auxiliary operations meant to guarantee the speed for the future.

Not so long ago an algorithmic idea of that class has appeared in the field of computer vision and works remarkably well — namely, the idea of Haar-like features due to Viola and Jones (2001, 2004) \cite{9,10}. Haar-like features are now commonly applied to detect objects (faces, people, vehicles, road signs, etc.) in digital...
images [1, 8]. One should be aware that the fast performance of Haar-like features is not owed to the nature of these features as such; they are simple differential features that can be viewed as rough contours (e.g. difference in average pixel intensity between forehead and eyes regions). Instead, the fast performance is in fact a consequence of a computational trick known as integral image. For an image \(i(x, y)\) the elementary integral image is:

\[
ii(x, y) = \sum_{1 \leq j \leq x} \sum_{1 \leq k \leq y} i(j, k).
\]

Once such a cumulant is prepared, the sum of intensities over any image window can be calculated in constant time — \(O(1)\) — regardless of the number of pixels, using 2 subtractions and 1 addition. This allows for very fast feature extraction.

There exist a few modifications of that idea. For example, a cumulant of squares \(ii(x, y) = \sum_{1 \leq j \leq x} \sum_{1 \leq k \leq y} i(j, k)^2\) is useful for calculations of variance. In turn, cumulants of so-called vote matrices allow for extraction of HOG\(^2\) features [5, 7]. Yet, other propositions of that kind are scarce and, in generality, approaches which would allow for constant-time extraction of more advanced features, exhibiting better approximation properties, are not known.

In this paper we demonstrate that it is possible to prepare a set of cumulants of form:

\[
ii(x, y) = \sum_{1 \leq j \leq x} \sum_{1 \leq k \leq y} i(j, k) \cdot \cos f(j, k, \cdots) \quad \text{and} \quad ii(x, y) = \sum_{1 \leq j \leq x} \sum_{1 \leq k \leq y} i(j, k) \cdot \sin f(j, k, \cdots),
\]

with \(f\) being a suitably chosen function, and then to use the cumulants to extract Fourier moments of low orders in constant-time, using 21 operations, regardless of size and position of detection window.

We omit the topic of classifiers cascade in the paper.

\[\text{Hist}\]ogram of Oriented Gradients.

2 Haar-Like Features — Short Review

In this section we briefly remind Haar-like features and point out their connection to Haar wavelets.

Recall the mother Haar wavelet \(\psi(x)\) defined to yield: 1 for \(0 \leq x < \frac{1}{2}\), −1 for \(\frac{1}{2} \leq x < 1\), and 0 otherwise. The descendant wavelets are generated as follows:

\[
\psi_{j,k}(x) = \psi(2^{j-1}x - k), \quad j = 2, 3, \ldots; \quad k = 0, 1, \ldots 2^{j-1} - 1.
\]

Thus, descendants are narrowed and shifted versions of the mother wavelet. For any continuous function \(f\) (to be approximated) the orthogonality of wavelets

\[
\forall (j, k) \neq (l, m) \quad \langle \psi_{j,k}, \psi_{l,m} \rangle = \int_0^1 \psi_{j,k}(x) \psi_{l,m}(x) \, dx = 0,
\]

allows to write down the following expansion

\[
f(x) = c_0 \cdot 1 + \sum_{j=1}^{\infty} \sum_{k=0}^{2^{j-1}-1} c_{j,k} \psi_{j,k}(x),
\]

where the best coefficients can be found through inner products of Haar bases and the target function: \(c_{j,k} = 1/\|\psi_{j,k}\|^2 \langle f, \psi_{j,k} \rangle\) and \(c_0 = \langle f, 1 \rangle\).
Viola and Jones [9,10] proposed two-dimensional templates resembling Haar wavelets. The templates can be mapped to features by anchoring them within an image window at different positions and scales, and then calculating the difference in average intensity of pixels under white (+1) and black (−1) regions. We depict the templates and their connection to wavelets in Fig. 1. The intention of Viola and Jones was to generate a massive multitude of features (e.g. ∼ 10^5), so that some of them might happen to represent good characteristics of target objects (e.g. for faces: differences between forehead and eyes, nose and cheeks, etc.). Therefore, the way to implement how Haar-like features are actually embedded inside a window (i.e. setting up their positions and scales) is fairly arbitrary. One may allow for overlapping of feature supports and neglect orthogonality. On the other hand, we remark that it is straightforward to define orthogonal two-dimensional wavelets via products $\psi_{j,k,l,m}(x, y) = \psi_{j,k}(x) \cdot \psi_{l,m}(y)$, and to write down a polynomial in wavelets to approximate some fragment of image function $i(x, y)$. Note that in the formula for coefficients, $c_{j,k,l,m} = 1/\|\psi_{j,k,l,m}\|^2 \langle i, \psi_{j,k,l,m} \rangle$, the expression $\langle i, \psi_{j,k,l,m} \rangle$ is then equivalent to taking the white-black difference, as in the definition of Haar-like features, whereas the normalization constant $1/\|\psi_{j,k,l,m}\|^2$ plays the role of averaging (provided that white and black supports are of the same size).

3 Constant-Time Fourier Moments via Integral Images

Consider the following approximation, by a partial Fourier sum, of an image fragment restricted to a rectangle spanning from $(x_1, y_1)$ to $(x_2, y_2)$:

$$i(x, y) \approx \sum_{-n \leq k_x \leq n} \sum_{-n \leq k_y \leq n} c_{x_1,y_1} c_{x_2,y_2} 2\pi i \left( k_x \frac{x-x_1}{N_x} + k_y \frac{y-y_1}{N_y} \right), \quad x_1 \leq x \leq x_2, \quad y_1 \leq y \leq y_2, \quad (4)$$

where: $n$ is the harmonic order of approximation (variable-wise), $i = \sqrt{-1}$ is the imaginary unit (please note the calligraphic difference from $i$ denoting the image), the coefficients $c$ are complex numbers, and $N_x=x_2-x_1+1$, $N_y=y_2-y_1+1$. 

Fig. 1. Five 2D templates for Haar-like features defined in terms of 1D wavelets.
\[ N_y = y_2 - y_1 + 1 \] are rectangle widths in pixels. The superscripts \( k_x, k_y \) of \( c \) coefficients indicate the particular harmonic indexes. The subscripts represent the boundaries of the rectangle. In the current context the boundaries are fixed, but shall vary later when partitioning of the detection window becomes involved. To avoid confusion, we explain that throughout the paper \( N_x, k_x \) and similar subscript notations should not be treated as functions of the specific subscript value, but instead as an indication of what coordinate the quantity is associated with.

Due to orthogonality of Fourier bases, the optimal complex coefficients from (4) can be derived as

\[
c_{k_x, k_y}^{x_1, y_1} = \frac{1}{N_x N_y} \sum_{x_1 \leq x \leq x_2} \sum_{y_1 \leq y \leq y_2} i(x, y) e^{-2\pi i \left( \frac{k_x x - x_1 N_x + k_y y - y_1 N_y}{N_x N_y} \right)}.
\] (5)

From now on, we shall refer to the coefficients as Fourier moments, and we intend to use their real and imaginary parts as features for learning and detection.

Let us introduce two sets of integral images:

\[
\{ i_{i\cos}^{k_x, k_y} \}_{N_x, N_y}, \{ i_{i\sin}^{k_x, k_y} \}_{N_x, N_y},
\]

related to cosine and sine functions, respectively, and constructed as follows:

\[
i_{i\cos}^{k_x, k_y} (x, y) = \sum_{1 \leq j_x \leq x} \sum_{1 \leq j_y \leq y} i(j_x, j_y) \cos \left( -2\pi \left( \frac{k_x j_x}{N_x} + \frac{k_y j_y}{N_y} \right) \right), \quad (6)
\]

\[
i_{i\sin}^{k_x, k_y} (x, y) = \sum_{1 \leq j_x \leq x} \sum_{1 \leq j_y \leq y} i(j_x, j_y) \sin \left( -2\pi \left( \frac{k_x j_x}{N_x} + \frac{k_y j_y}{N_y} \right) \right), \quad (7)
\]

where indexes \((k_x, k_y)\) iterate over the set:

\[
\{ (k_x, k_y) : -n \leq k_x \leq -1, -n \leq k_y \leq n \} \cup \{ (0, k_y) : -n \leq k_y \leq -1 \} \cup \{ (0, 0) \}.
\] (8)

We remark that a single integral image of form (6) or (7) can be calculated by induction in linear time with respect to the total number of pixels in the input image (i.e. with one pass).

Let us now define the growth operator for any integral image \( i\) taken from either of the sets \( \{ i_{i\cos} \}, \{ i_{i\sin} \} \):

\[
\Delta_{x_1, y_1}^{x_2, y_2} (i) = i(x_2, y_2) - i(x_1 - 1, y_2) - i(x_2, y_1 - 1) + i(x_1 - 1, y_1 - 1).
\] (9)

Note that \( \Delta \) returns a subsum over given cuboid in constant time using 2 subtractions and 1 addition, instead of \( \Theta(N_x N_y) \) operations.

The following proposition constitutes the main contribution of the paper.

**Proposition 1.** Suppose the two sets of integral images:

\[
\{ i_{i\cos}^{k_x, k_y} \}_{N_x, N_y}, \{ i_{i\sin}^{k_x, k_y} \}_{N_x, N_y},
\]
defined as in (6) and (7), respectively, have been calculated prior to the detection procedure. Then, for any rectangle of widths $N_x$, $N_y$ in the image, the real and imaginary parts of each of its Fourier moments can be calculated in constant time — $O(1)$ — as follows:

$$
\text{Re} \left( \frac{k_x, k_y}{c_{x_1, y_1}^{x_2, y_2}} \right) = \frac{1}{N_x N_y} \left( \cos \left( 2\pi \left( \frac{k_x x_1}{N_x} + \frac{k_y y_1}{N_y} \right) \right) \Delta_{x_1, y_1}^{x_2, y_2} \left( i \frac{k_x, k_y}{N_x, N_y} \right) 
- \sin \left( 2\pi \left( \frac{k_x x_1}{N_x} + \frac{k_y y_1}{N_y} \right) \right) \Delta_{x_1, y_1}^{x_2, y_2} \left( i \frac{k_x, k_y}{N_x, N_y} \right) \right), \quad (10)
$$

$$
\text{Im} \left( \frac{k_x, k_y}{c_{x_1, y_1}^{x_2, y_2}} \right) = \frac{1}{N_x N_y} \left( \sin \left( 2\pi \left( \frac{k_x x_1}{N_x} + \frac{k_y y_1}{N_y} \right) \right) \Delta_{x_1, y_1}^{x_2, y_2} \left( i \frac{k_x, k_y}{N_x, N_y} \right) 
+ \cos \left( 2\pi \left( \frac{k_x x_1}{N_x} + \frac{k_y y_1}{N_y} \right) \right) \Delta_{x_1, y_1}^{x_2, y_2} \left( i \frac{k_x, k_y}{N_x, N_y} \right) \right). \quad (11)
$$

As one can note both parts, real (10) and imaginary (11), require a calculation of two growth operations and two trigonometric functions. It is easy check that this comprises a total of 21 operations: 8 additions (or subtractions), 8 multiplications, 3 divisions, and 2 trigonometric functions for either of the two formulas. Note that it is sufficient to calculate the argument under trigonometric functions only once. Furthermore, it is worth noting that this argument depends on the offset $(x_1, y_1)$ of the rectangle, but does not depend on the rectangle contents — the pixels, thereby making the overall calculation a constant-time calculation. The proof of the proposition is a straightforward derivation.

**Proof.** Rewriting the moments from (5) using Euler’s identity leads to:

$$
k_x, k_y \frac{c_{x_1, y_1}^{x_2, y_2}}{x_2, y_2} = \frac{1}{N_x N_y} \sum_{x_1 \leq x \leq x_2} \sum_{y_1 \leq y \leq y_2} i(x, y) \left( \cos \left( -2\pi \left( \frac{k_x x - x_1}{N_x} + \frac{k_y y - y_1}{N_y} \right) \right) 
+ i \sin \left( -2\pi \left( \frac{k_x x - x_1}{N_x} + \frac{k_y y - y_1}{N_y} \right) \right) \right). \quad (12)
$$

The argument of the trigonometric functions can be parted into a group of terms independent from the pixel index $(x, y)$ and a group dependent on it as follows:

$$
\alpha = 2\pi \left( \frac{k_x x_1}{N_x} + \frac{k_y y_1}{N_y} \right), \\
\beta(x, y) = -2\pi \left( \frac{k_x x}{N_x} + \frac{k_y y}{N_y} \right).
$$

Now, one can apply in (12) the trigonometric identities for $\cos(\alpha + \beta)$ and $\sin(\alpha + \beta)$. Simultaneously, the $\cos \alpha$ and $\sin \alpha$ terms can be pulled out as factors in front of the summations as they are independent of the pixel index $(x, y)$. Finally, by splitting the expression into real and imaginary parts one obtains:
The underbraces show how the expensive summations over pixels get replaced by cheap (constant-time) growths of integral images, yielding (10), (11).

The form of indexes set (8) is implied by the known symmetry property i.e. complex conjugacy of opposed Fourier coefficients:

\[
\text{Re}\left(\frac{c_{x_1,y_1}}{x_2,y_2}^{k_x,k_y}\right) = \text{Re}\left(\frac{c_{x_1,y_1}}{x_2,y_2}^{-k_x,-k_y}\right), \quad \text{Im}\left(\frac{c_{x_1,y_1}}{x_2,y_2}^{k_x,k_y}\right) = -\text{Im}\left(\frac{c_{x_1,y_1}}{x_2,y_2}^{-k_x,-k_y}\right),
\]

and also by the fact that the zeroth order moment is a real number — \(\text{Im}(c^{0,0}) = 0\). Hence, it suffices to calculate roughly only a half of all moments. More precisely, the effective number of distinct moments is

\[
1/2 ((2n + 1)^2 - 1) + 1,
\]

which yields \(2n^2 + 2n + 1\) and corresponds to the size of set (8). In fact, any set of \(2n^2 + 2n + 1\) coefficients will do to uniquely reconstruct all coefficients.

As regards the needed number of integral images, it is equal to the double of expression (15) yielding: \((2n + 1)^2 + 1\), since required are two kinds of integral images, related to cosine and sine functions, for each \((k_x, k_y)\) pair. Hence, the calculation of all cumulants is potentially expensive. That is why, when using Proposition 1, one should in practice limit himself to low harmonic orders, so that the time invested in the preparation of integral images is reasonably small.

### 4 Window Partitioning — Piecewise Approximations

Apart from \(n\) let us now introduce an additional integer parameter \(p > 0\), responsible for the partitioning of detection window and affecting the final number of features. Let the window be partitioned into a regular grid of rectangles: \(p \times p\). The moments shall be extracted from each rectangle independently and their concatenation shall form the final vector of features. This approach can be understood as a piecewise Fourier approximation of the window under detection.

Consider a single image pass with a detection window of size \(w_x \times w_y\). The partitioning leads to a grid of pieces with widths equal to: \(N_x = \lfloor w_x/p \rfloor\),...
$N_y = \lfloor w_y/p \rfloor$. We denote the corresponding division remainders as: $m_x = w_x \mod p$, $m_y = w_y \mod p$. Now, for a window starting at a point $(x_1, y_1)$ and for fixed numbers $N_x, N_y$ we define the collection of features $\{ f_{x_1,y_1}^{N_x,N_y}(\cdots) \}$ as follows:

$$f_{x_1,y_1}^{N_x,N_y}(k_x,k_y,p_x,p_y,r) = \begin{cases} 
\text{Re}\left( \frac{c_{k_x,k_y}}{c_{x_1+p_xN_x,y_1+p_yN_y}} \right), & r = 1; \\
\text{Im}\left( \frac{c_{k_x,k_y}}{c_{x_1+p_xN_x,y_1+p_yN_y}} \right), & r = 0;
\end{cases} \quad (16)$$

where: $(x_1', y_1') = (x_1 + \lfloor m_x/2 \rfloor, y_1 + \lfloor m_y/2 \rfloor)$ represents a shifted starting point taking into account small corrections due to the partitioning remainders; indexes $k_x, k_y$ iterate over the set defined in (8); $0 \leq p_x, p_y \leq p-1$ represent the index (and hence the offset) of a particular rectangle; and $r$ is a flag switching between real and imaginary parts. Since $\text{Im}(c^{0,0}) = 0$, then $f(0,0, p_x, p_y, 0)$ is also zero for any $p_x, p_y$ pair, and therefore should not be taken as an actual feature. Finally, the total number of features is: $d(n,p) = (2n+1)^2p^2$.

Figures 2 and 3 show example reconstructions of an image from Fourier moments. Reconstructions are carried out according to formula (4) (piecewise reconstructions for $p > 1$). Obviously, image reconstruction as such is not a needed step in a detection procedure. Yet, the quality of reconstructions helps to understand the descriptive capability of the features. Under each reconstruction

---

3 This operation centers the grid of rectangles within the scanning window.
we report the mean absolute error (MAE) and the ratio of the number of features to the number of pixels (feats/pxs).

5 Face Detection Experiments

Taking advantage of Proposition 1, we have trained four variants of face detectors based on Fourier moments. The variants correspond to different settings of $n$ and $p$ parameters: (1) $n = 2, p = 5$ (625 features), (2) $n = 2, p = 7$ (1 225 features), (3) $n = 3, p = 5$ (1 225 features), (4) $n = 3, p = 7$ (2 401 features).

To compare accuracy, we have introduced opponents for our Fourier-based detectors — namely, additional detectors trained on the same learning material but using Haar-like features. Our intention was to impose a similar feature space parameterization in both approaches, but to slightly favour the Haar-like features in terms of their quantity. To achieve this, we were using 5 Haar templates discussed earlier (Fig. 1) and we were anchoring the Haar-like features within the detection window on $p \times p$ grids $(p = 5, 7)$ — hence, the grids were of the same sizes as in the case of Fourier moments. Lengths of Haar-like features were scaled independently along each axis, and the number of scales was controlled by an additional parameter $q > 0$. More precisely, for a window of size $w_x \times w_y$, the lengths of features were changing according to: $w_x \lambda^{s_x}$ and $w_y \lambda^{s_y}$ with $1 \leq s_x, s_y \leq q$ and the scaling factor chosen to be $\lambda = \sqrt{2}/2$. Hence, the final number of generated Haar-like features became $d_{HF}(q, p) = 5q^2p^2$. We remind that the corresponding number for Fourier moments is $d_{FM}(n, p) = (2n + 1)^2p^2$.

A learning material of moderately large size was used. It contained 7 258 positive examples (windows with faces marked manually from 3 000 images) and 100 000 negative examples (windows sampled randomly from non-face images). Accuracy measures were evaluated on test data consisting of 500 images with 1 000 faces and a total of 70 252 859 windows. In order to produce ROC curves for detectors on the basis of test material, a test set with a limited number of negatives was randomly selected (to fit in RAM memory). We imposed $2 \cdot 10^6$ negative windows in that set, thereby making the precision along the FAR$^4$ axis at the level of $5 \cdot 10^{-7}$. Details of the experimental setup are listed in Table 1.

We have applied a boosted learning algorithm known as RealBoost+bins, see e.g. [6], with additional weight trimming [3]. In this variant, an ensemble consists of partial (weak) classifiers that are based on selected single features each. Classifiers’ responses are real-valued, equal to half the logit transform, and a binning mechanism is introduced to store those responses. We have set up 8 bins of equal widths per feature. Finally, $T = 256$ or $T = 512$ rounds of boosting were carried out, yielding ensembles with at most $T$ distinct features selected.

The software has been programmed in C#, with key procedures (e.g. integral images, features extraction) implemented for efficiency in C++ as dll libraries.

We start the review of results by showing in Fig. 4 some example outcomes produced by a Fourier-based detector (variant: $n = 3, p = 7$). The left-hand

\footnote{False Alarm Rate.}
Table 1. Setup for face detection experiments.

| Train data                      | Value   | Additional information                                      |
|---------------------------------|---------|-------------------------------------------------------------|
| No. of images with faces       | 3 000   | Photos downloaded from Google Images for queries: person, people, group of people, family, children, sportsmen, students, etc. |
| No. of images without faces    | 300     | As above, queries: view, landscape, street, cars, etc.      |
| No. of positive examples       | 7 258   | Face windows marked manually                                |
| No. of negative examples       | 100 000 | Imposed quantity; examples sampled at random positions and scales within images without faces |
| Train set size                 | 107 258 | Positive and negative examples in total                     |

| Test data                      | Value   | Additional information                                      |
|--------------------------------|---------|-------------------------------------------------------------|
| No. of images with faces       | 500     | Queries as for train data (other images)                    |
| No. of images without faces    | 300     | Queries as for train data (other images)                    |
| No. of positive examples       | 1 000   | Face windows marked manually                                |
| No. of negative examples       | 2 000 000 | Imposed quantity; examples sampled at random              |
| Test set size                  | 2 001 000 | Positive and negative windows in total                   |

Detection procedure (scanning with a sliding window)

| Parameter                      | Value   | Description                                                      |
|--------------------------------|---------|------------------------------------------------------------------|
| Image height                   | 480     | Before detection, images scaled to the height 480, keeping original height: width proportion |
| No. of detection scales        | 8       | Images scanned with 8 different sizes of window                  |
| Window growing coefficient     | 1.2     | Window widths and heights increase by ≈20% per scale             |
| Smallest window size           | 48 × 48 | Faces smaller than ≈10% of image height not to be detected       |
| Largest window size            | 172 × 172 | Faces larger than ≈36% of image height not to be detected       |
| Window jumping coefficient     | 0.05    | Window jumps equal to ≈5% of its width and height               |

side images contain all single positive indications. Their counterparts on the right-hand side are postprocessed outcomes, i.e. after grouping of windows clusters has been performed. Figure 5 shows some examples of false alarms, interesting because of their resemblance to faces (we encourage to zoom the document).

ROC curves for all detectors are presented in Fig. 6. To distinguish the curves better, logarithmic scale was imposed on FAR axis. Operational decision
thresholds for detectors were taken as averages of threshold values registered for two left-most points on ROCs, with the smallest FAR values ($\approx 5 \cdot 10^{-7}$).

Detailed accuracy results are reported in Table 2. In particular, AUC (area under ROC) measures are stated. In learning tasks with strongly imbalanced classes, it is the left-most part of the ROC curve that is of crucial importance. Therefore, we decided to report normalized AUCs obtained up to several first orders of magnitudes of FAR. More precisely, $\text{AUC}_\alpha$ should be understood as $1/\alpha \int_0^\alpha s(f) df$ where $s$ and $f$ represent sensitivity and FAR respectively.
Fig. 6. Comparison of ROC curves for detectors based on Fourier moments (FM, black) and Haar-like features (HF, gray). Logarithmic scale imposed on FAR axis.

Table 2. Accuracy measures of detectors on test data (best variant in gray).

| name / description | $\text{AUC}_{\alpha}$ | sensitivity | FAR per images | FAR per windows | accuracy per windows |
|--------------------|------------------------|-------------|----------------|-----------------|---------------------|
| $HF \ q=3, p=5 \ (1125); \ T=512$ | $0.6761, 0.8123, 0.9156, 0.699, 0.098, 6.975 \cdot 10^{-7}$ | $0.999995018084708$ |
| $FM \ n=2, p=5 \ (625); \ T=512$ | $0.8401, 0.9236, 0.9714, 0.889, 0.092, 6.548 \cdot 10^{-7}$ | $0.999997765249097$ |
| $HF \ q=4, p=5 \ (2000); \ T=512$ | $0.8021, 0.9082, 0.9624, 0.849, 0.086, 6.121 \cdot 10^{-7}$ | $0.999997238589628$ |
| $FM \ n=3, p=5 \ (1225); \ T=512$ | $0.8475, 0.9285, 0.9703, 0.872, 0.054, 3.843 \cdot 10^{-10}$ | $0.999997793717795$ |
| $HF \ q=3, p=7 \ (2205); \ T=512$ | $0.7075, 0.8376, 0.9320, 0.741, 0.084, 5.978 \cdot 10^{-7}$ | $0.999995715550288$ |
| $FM \ n=2, p=7 \ (1225); \ T=512$ | $0.8800, 0.9480, 0.9826, 0.924, 0.058, 4.128 \cdot 10^{-10}$ | $0.999998505420626$ |
| $HF \ q=4, p=7 \ (3920); \ T=512$ | $0.8188, 0.9141, 0.9729, 0.857, 0.052, 3.701 \cdot 10^{-7}$ | $0.999997594441206$ |
| $FM \ n=3, p=7 \ (2401); \ T=512$ | $0.8965, 0.9538, 0.9845, 0.945, 0.052, 3.701 \cdot 10^{-7}$ | $0.999998847038424$ |

Test results clearly show that detectors based on Fourier moments surpass in accuracy their counterparts based on Haar-like features, even though there were fewer features at disposal at the learning stage. It is an experimental evidence that Fourier moments have better approximation properties for the face detection problem. Naturally, the best (and definitely satisfactory) accuracy was achieved by the variant with the most rich space consisting of 2,401 features ($n = 3, p = 7$). Please note that the number $2.4 \cdot 10^3$ is decidedly smaller than the total of $1.8 \cdot 10^5$ features originally used by Viola and Jones in their experiment [9].
Table 3. Time performance for a 480 × 480 image (parallel computations on: Intel i7 Q 720 4×2-core 1.60 GHz CPU).

| Quantity (or operations)                  | Fourier moments ($T = 512$) | Haar-like features ($T = 512$) |
|------------------------------------------|-------------------------------|---------------------------------|
|                                          | (456 distinct feats.)         | (472 distinct feats.)           |
| No. of analyzed windows                  | 108 873                       | 108 873                         |
| No. of prepared integral images          | 400                           | 1                               |
|                                          | (50 images per each of 8 scales) |                                |
| Preparation time for integral images     | 1 015 ms                      | 9 ms                            |
| Preparation time per 1 integral image    | 2.54 ms                       | 9 ms                            |
| Total time of detection procedure        | 2 852 ms                      | 1 032 ms                        |
| Time per 1 window                        | 27.20 µs (amortized: 16.87 µs) | 9.48 µs                         |
| Time per 1 window and 1 feature          | 57.45 ns (amortized: 37.00 ns) | 20.08 ns                        |

Finally, Table 3 reports the time performance we achieved on our machine (Intel i7 Q 720 4×2-core 1.60 GHz CPU). Please remember that cascades of classifiers were purposely not involved in the experiment. The observed amortized extraction time for a single Fourier feature was approximately 37 ns — about two times longer than for a Haar-like feature. This is related to the 21 operations we declared in Proposition 1, and is roughly proportional to the number of operations needed for Haar-like features (8 operations for an ‘edge’ template, 12 for a ‘diagonal’ template). We remark that by involving a cascade of classifiers and a processor with more cores/threads (to save time for preparation of integral images) the real-time regime could be achieved without difficulty in both approaches (FM and HF).

6 Conclusions

We have proposed a computational technique, based on special integral images, for constant-time extraction of low order Fourier moments. The technique is suitable for detection tasks. Our experiments have shown that fairly small sets of Fourier features — real and imaginary parts of moments — can lead to face detectors superior in accuracy than Haar-based detectors. The proposed approach could be beneficial in other machine learning applications where accuracy is of primary importance (e.g. medical diagnosis, image-based fault detection in production, landmine detection [4,5]), and where one is willing to invest some additional time in the preparation of special integral images in order to improve accuracy.
References

1. Charles, J., et al.: Automatic and efficient human pose estimation for sign language videos. Int. J. Comput. Vis. 110(1), 70–90 (2014)
2. Cormen, T.H., et al.: Introduction to Algorithms, 3rd edn. MIT Press, Cambridge (2009)
3. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Stat. 28(2), 337–407 (2000)
4. Klóesk, P., Kapruziak, M., Olech, B.: Fast Extraction of 3D Fourier Moments via Multiple Integral Images: An Application to Antitank Mine Detection in GPR C-scans. In: Chmielewski, L.J., Datta, A., Kozera, R., Wojciechowski, K. (eds.) ICCVG 2016. LNCS, vol. 9972, pp. 206–220. Springer, Cham (2016). doi:10.1007/978-3-319-46418-3_19
5. Klóesk, P., Godziuk, A., Kapruziak, M., Olech, B.: Fast Analysis of C-scans from Ground Penetrating Radar via 3D Haar-like Features with Application to Landmine Detection. IEEE Trans. Geosci. Remote Sens. 53(7), 3996–4009 (2015)
6. Rasolzadeh, B., et al.: Response binning: improved weak classifiers for boosting. In: IEEE Intelligent Vehicles Symposium, pp. 344–349 (2006)
7. Said, Y., Atri, M., Tourki, R.: Human detection based on integral Histograms of Oriented Gradients and SVM. In: Communications, Computing and Control Applications (CCCA 2011), pp. 1–5. IEEE (2011)
8. Tresadern, P.A., Ionita, M.C., Cootes, T.F.: Real-Time Facial Feature Tracking on a Mobile Device. Int. J. Comput. Vis. 96(3), 280–289 (2012)
9. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Conference on Computer Vision and Pattern Recognition (CVPR 2001), pp. 511–518. IEEE (2001)
10. Viola, P., Jones, M.: Robust Real-time Face Detection. Int. J. Comput. Vis. 57(2), 137–154 (2004)
Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.