Fast QR Code Detection in Arbitrarily Acquired Images

Luiz F. F. Belussi and Nina S. T. Hirata
Department of Computer Science, Institute of Mathematics and Statistics,
University of São Paulo — São Paulo, Brazil
Email: belussi@ime.usp.br , nina@ime.usp.br

***Abstract*—The detection of QR codes, a type of 2D barcode, as described in the literature consists merely in the determination of the boundaries of the symbol region in images obtained with the specific intent of highlighting the symbol. However, many important applications such as those related with accessibility technologies or robotics, depends on first detecting the presence of a barcode in an environment. We employ Viola-Jones rapid object detection framework to address the problem of finding QR codes in arbitrarily acquired images. This framework provides an efficient way to focus the detection process in promising regions of the image and a very fast feature calculation approach for pattern classification. An extensive study of variations in the parameters of the framework for detecting finder patterns, present in three corners of every QR code, was carried out. Detection accuracy superior to 90%, with controlled number of false positives, is achieved. We also propose a post-processing algorithm that aggregates the results of the first step and decides if the detected finder patterns are part of QR code symbols. This two-step processing is done in real time.**

***Keywords*-QR code; 2D barcode; Haar-like features; cascade classifier; boosting; classification; pattern recognition;**

I. Introduction

QR codes (ISO/IEC 18004) are the type of 2D barcode with the sharpest increase in utilization in the last years. Fig. 1 shows some examples of QR code symbols. The most common uses of QR codes are as "physical hyperlinks" connecting places and objects to websites. QR codes have been designed to be easily found and to have its size and orientation determined under bad imaging conditions. In addition, ISO/IEC 18004 specifies an error correction scheme that can recover at most 30% of occluded or damaged symbol area. These features make the QR code an extremely well succeeded technology in the area of 2D barcodes.

Fig. 1. Examples of QR code symbols.

Many applications are related to aiding visually impaired people [1], [2], [3] or to QR code processing in video and mobile devices [4], [5], [6]. An approach to the detection of 2D barcodes in the PDF417 standard, to help visually impaired people identify objects using these barcodes as tags, is proposed in [1]. The use of QR codes in helping blind people identify objects is the problem addressed in [2]. In addition, it is not difficult to envisage applications in robot navigation.

The existing QR code decoders require the symbol to be properly "framed", that is, the symbol must correspond to at least 30% of the image area to be suitable for decoding. Often, even images acquired with the specific intent of capturing a QR code are subject to noise, blur, rotation, perspective distortion, uneven illumination, or partial code area occlusion. Decoders may fail due to such noise and distortions. Therefore, most works in related literature, concerned with barcode detection or recognition, deal with the problem of enhancing or processing these type of images in order to facilitate decoding [7].

Most works proposing the use of these codes to help visually impaired and blind people do not address the problem of getting a person with this kind of disability to correctly point the camera towards the barcode symbol [1], [2]. However, they do acknowledge that processing arbitrary acquired images for the detection of 2D barcodes and development of computationally efficient algorithms to recognize the barcode symbols in low light and low resolution situations are challenging and relevant issues [8]. To the best of our knowledge, there are only very few works dealing with the problem of detecting QR codes in arbitrarily acquired images. Furthermore, in such cases, detection is done with the help of visual cues (see an example in Fig. 2) or with the help of RFID tags [9].

The aim of this work is to develop a method for detecting QR codes in arbitrarily acquired images. Once the presence of a QR code is detected, the camera holder or a robot can be guided either vocally or by some appropriate set of commands to correctly frame the QR code. We are interested not only in

Fig. 2. Example of visual cue to aid QR code detection (reproduced from [9] with permission).

detecting the presence of a code, but also in determining the size and position of a QR code symbol in an image, important for correctly framing them. In addition, detection should be fast to allow real-time applications.

A promising approach to address this problem is the rapid object detection framework, proposed by Viola and Jones, and originally applied for face detection [10]. In their framework, detection is based on a cascade of weak classifiers with simple features inspired by Haar wavelets. They achieve an impressive speed in multi-scale face detection, combining simplicity in feature computation with an effective way of focusing processing only in promising regions of the image.

Specifically, in this work we investigate the use of Viola-Jones framework for the detection of finder patterns (FIPs), a fixed pattern located in three corners of any QR code. Since the training framework has many parameters that can be tuned to meet the needs of a particular application, we investigate parameter variation effects in several experimental scenarios. Results are evaluated in terms of hits (true positives) and false alarms (false positives) and they show that a relatively small number of FIP samples in the training set is sufficient to achieve very good detection rates. Furthermore, since the detection step typically presents false alarms, a geometrical constraint based post-processing step is proposed in order to decide if the detected FIPs are or not corners of a QR code symbol.

The rest of this paper is organized as follows. In Section II, the general structure of a QR code is described and Viola-Jones rapid object detection framework is reviewed. In Section III, the proposed method for QR code detection is described. We first justify the decision on detecting FIPs rather than the whole QR code area, then we describe the training and classifier evaluation procedure, which is followed by the description and discussion of an approach for choosing parameter values. In Section IV we describe and discuss the main experimental results with respect to parameter choice. We also show examples of QR code symbol detection in real images. Finally, in Section V, we present a summary of the main results and some steps for future work.

II. Background

A. QR code structure

The ISO/IEC standard 18004 defines the QR code symbol as having a general structure that comprises, besides data, version information, and error correction code words, the following regions:

- a quiet zone around the symbol
- 3 finder patterns (FIP) in the corners
- 2 timing patterns (TP) between the finder patterns
- N alignment patterns (AP) inside the data area

as illustrated in Fig 3.

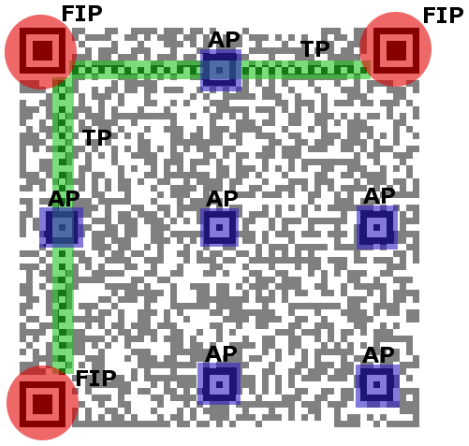


Fig. 3. QR code structure (FIP: Finder Pattern / TP: Timing Pattern / AP: Alignment Pattern).

The finder patterns are specially designed to be found in any search direction as the sequence of black (b) and white (w) pixels along any scan line that passes through its center preserve the special sequence and size ratio w:b:w:bbb:w:b:w, as can be seen in Fig. 4.

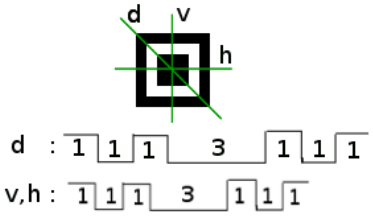
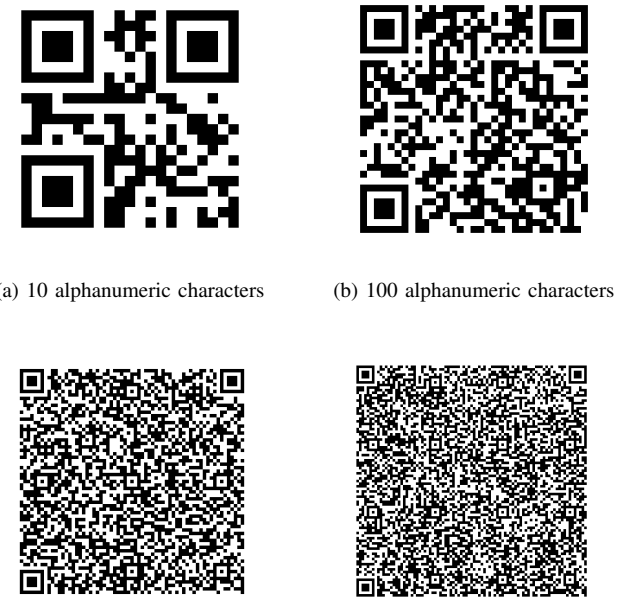


Fig. 4. Black and white pixel proportions in finder patterns for diagonal (d), vertical (v) and horizontal (h) scan lines.

Different instances of QR codes may have different internal proportions with respect of its structural components depending on the amount of data encoded. This amount determines an appropriate "version" of QR code that must be chosen, that is, there are various predefined sizes of symbols (with respect to the number of modules in symbol area, called versions) with distinct information storage capacity that may be used in QR code generation. Timing and alignment patterns are used to control the internal structure (see Fig. 5).



(a) 10 alphanumeric characters (b) 100 alphanumeric characters

(c) 500 alphanumeric characters (d) 1000 alphanumeric characters

Fig. 5. Different versions of QR code symbols.

B. Viola-Jones' object detection framework

Viola and Jones [10] proposed a boosted cascade of simple classifiers for rapid object detection in images. Classifier efficiency is a consequence of using simple Haar-like features that can be computed in constant time using the so called integral images. Cascading and boosting provide effective aggregation of weak classifiers and allow, to some extent, control of hit and false alarm rates in every stage of the resulting classifier.

1) Haar-like Features: The features used by the classifiers proposed in [10] are based on feature prototypes, inspired from [11], and shown in Fig. 6. They are known as Haar-like features due to their similarity to rescaled "square-shaped" functions that form the basis of the Haar wavelet family. An

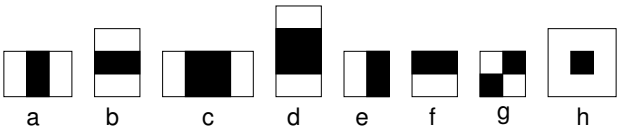


Fig. 6. Basic feature prototype set.

extension of this basic set, including also the 45 degree rotated prototypes, was proposed in [12] (see Fig. 7).

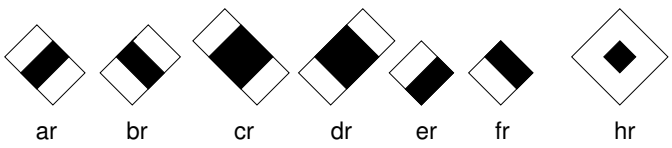
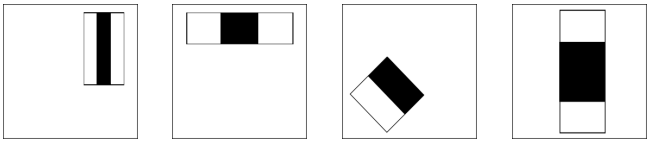


Fig. 7. Additional rotated feature prototypes in the extended set.

Feature values are computed with respect to a sample that corresponds to a sub-region of the image under a sliding evaluation window. A feature is defined as an instance of some prototype when its width, height and relative position in the evaluation window are defined, as in Fig. 8. By definition, the value of a feature is the sum of the pixel values in the white rectangle area subtracted from the corresponding summation in the black rectangle area.



(a) Based on prototype "a" (b) Based on prototype "a" (c) Based on prototype "fr" (d) Based on prototype "d"

Fig. 8. Examples of Haar-like features.

2) Integral Image: The integral image [10] is a matrix with the same dimensions of the input image in which every (x, y) position stores the sum of all pixel values in the rectangle between $(0, 0)$ and (x, y). Given an image i of dimension $h \times w$, the calculation of the corresponding integral image ii can be performed in approximately $2*(w*h)$ memory accesses based on the following recurrence:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2)$$

where $s(x, y)$ is the cumulative row sum, with $s(x, -1) = 0$, and $ii(-1, y) = 0$.

Once the integral image is calculated, the sum of the pixel values in any rectangle within the image can be determined in exactly 4 memory accesses, as shown in Fig. 9. Thus, feature values of the type above can be computed in constant time. For the extended feature prototype set, similar fast calculation

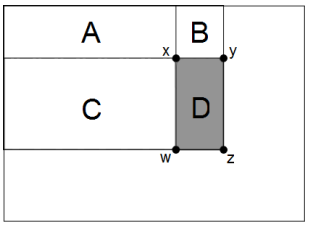


Fig. 9. The sum of pixels in the shaded area can be computed as $ii(z) - ii(y) - ii(w) + ii(x)$.

is possible using triangles instead of rectangles [12].

3) Classifier Cascade and Boosting: A cascade consists of a series of consecutive stages trained to reject samples that do not match the searched pattern while retaining the majority of positive samples and passing them to the next stage. A sample is said to be detected when it passes through the last cascade stage without rejection.

A sample consists of a sub-region of the image restricted to a window and the complete sample set of the image is

obtained by sliding the window on the image and varying its size from a given minimum to the largest possible window in the image. The classifier is applied to each element of the complete sample set.

Each stage is made from a set of weak classifiers aggregated in a committee by a boosting algorithm and can be seen as an independent classifier designed to obtain a very high hit rate (typically 99% or more) with an acceptable false alarm rate (typically between 40% and 60%). Every weak classifier is typically the result of the computation of a Haar-like feature followed by a binary threshold, although they can have more sophisticated forms like simple trees as will be considered in Section IV.

Boosting works in rounds, iteratively training a set of weak classifiers while reweighting the training samples so that "hard" samples will have increased relative importance in the set. In each round the best weak classifier is selected to compose the resulting classifier. Thus, boosting can achieve the specified hit rate and false alarm rate as it increases the weak classifiers in the combination (as long as the features used have enough representational power).

C. Classifier performance measures

Detection of QR codes can be formulated as a binary classification problem. Given a classifier that labels pixels or regions of an image as QR codes or not, its performance can be evaluated in terms of some well-known measures.

Due to the nature of Viola-Jones framework, which detects regions as being positive or negative instances, an appropriate measure should take into consideration how many positive samples are actually detected (true positives or TP) and also how many negative samples are wrongly detected as positive (false positives or FP). In this work, the first one is expressed in terms of recall, the percentage of total positive samples (true positives plus false negatives) that are recognized as positive. The second one is expressed by the number of false alarms (number of negative samples that are classified as positives).

$$\begin{aligned} \text{Recall} &= \text{TP}/(\text{TP}+\text{FN}) \\ \text{False Alarm or False Positive} &= \text{FP} \end{aligned}$$

III. Proposed detection approach

The proposed approach consists of two main steps:

1) Detection of FIP candidates using Viola-Jones framework: in this step, high recall rate is desirable.
2) Post-processing consisting of analysis and aggregation of FIP candidates to determine the size and position of the barcode: the idea is to combine FIP candidates and check whether they form a triangle compatible with QR code structure.

All images are processed after converting them to gray-scale.

A. Why FIPs?

The code size and position determination according to the ISO/IEC 18004 corresponds to the specification of finder patterns in three corners of the symbol. To estimate symbol size and position, it is suggested that scan lines should be used. In fact, when the barcode is properly framed, one can suppose FIPs correspond to a relatively large area within the image and thus appropriate gaps between scan lines can be considered to speed up computation. However, for arbitrarily acquired images, FIP sizes are also arbitrary and one has to consider consecutive scan lines, almost without gaps, in order to not miss a FIP. Moreover, any deformation or noise may disrupt the sequence of white and black pixels w:b:w:bbb:w:b:w that characterize FIPs. The impact of such disruption is much more critical in low resolution images. Thus, scan line based approaches are impractical for the detection of QR codes in arbitrary images.

In Viola-Jones framework it is assumed that the pattern to be detected should possess a rigid structure, but no strong restrictions concerning pattern size is assumed. As described in Section II-A, QR codes when considered in its whole do not have a rigid structure. On the other hand, FIPs are present in all QR codes and their structure is rigid, making them an excellent target for detection through Viola-Jones approach.

B. Training parameters

Several parameters can be tuned in the classifier cascade design. The following parameters have been considered in our investigation:

- **Feature set** (MODE): Two possibilities are the basic set (Fig. 6) and the extended set (Fig. 7).
- **Symmetry** (SYM): When the target pattern is symmetrical, the training algorithm can be restricted to consider only half part of the positive samples in order to reduce processing time.
- **Classifier topology** (MTS): It is possible to allow splits in the cascade, turning it into a simple CART tree [13]. Empirically observed positive impact in cascade performance when allowing splits is reported in [14].
- **Weak classifier number of splits** (NS): A weak classifier in its simplest form is just a single Haar-like feature and a binary threshold that will be combined with other weak classifiers by the boosting algorithm to form a stage. It is possible to allow weak classifiers to learn more complex relationships in the training pattern by letting them to be not just a single feature but a simple CART tree with a limited small number of splits.
- **Maximum false alarm rate / Minimum hit rate** (FA/HR): Each cascade stage must comply with a maximum false alarm rate and with a minimum hit rate for the samples supplied to it in order to be considered finished. A too low false alarm rate requirement may cause the stage to become overly complex diminishing the benefits of the cascade approach. A very high minimum hit rate restriction may have the same effect.
- **Number of training samples** (SAMPLES): The number of training samples provided for the training algorithm.
- **Size of training samples** (SIZE): The size of the training samples. For face detection, it has been observed that 20×20 is an appropriate size [14].

C. Classifier training and evaluation

Training and evaluation implementations of Viola-Jones' classifier are available as `opencv-haartraining` [15] and `opencv-performance` [15] utilities, respectively, and both were used in this work.

In order to assess classifier performance, a base set of FIPs and background images was considered. A base set consisting of 380 FIP samples cropped from a variety of different images constituted the source of positive samples, while a set of 1500 background images (without QR codes) were considered as sources for the negative samples. The number of positive samples can be augmented by simulating new samples from those in the base set. In particular, we considered random (yet limited) intensity variation and rotation in x, y (perspective) and z (planar), using the `opencv-createsamples` utility [15].

For testing, positive samples were randomly scaled and positioned in a background image. The utility `opencv-createsamples` facilitates evaluation of classifiers by automatically generating the augmented positive sample set and also the test images.

The available positive samples and background images were divided into training and test sets. All positive samples for training were generated from 285 FIPs in the base set, applying transformations with intensity variation limited to ± 80 and rotation limited in x, y to $\pm 23^\circ$ and in z to $\pm 34^\circ$, while the test samples were generated from the remaining 95 FIPs applying the same variation in intensity and in x and y, and in z limited to $\pm 23^\circ$. The background images were split into two subsets of equal size (750 images each), being used respectively for training and testing.

Although the selected detection framework does not provide invariance to rotation, due to the fact that FIPs are symmetric, high hit rates are achieved even for rotated samples.

D. Parameter choice for the classifier design

The determination of parameter values to be used in the cascade training for FIP detection was done comparing the effect of individual parameter variation while maintaining all the other parameters fixed. Although this analysis cannot be considered exhaustive, it provides a very good indication of the parameter variation influence in final detection performance while avoiding the combinatorial explosion that would arise from experimenting all possibilities. This methodology for parameter choice was inspired from the analysis done in a similar fashion by Lienhart et al. for the domain of face detection [14].

Figure 10 shows the order in which the parameters were considered. At the beginning, to evaluate *Feature Set* (MODE), the other parameters have been fixed, based on empirical observations, to: SYM=N (Asymmetric), MTS=0 (cascade topology), NS=2, FA=0.5, SAMPLES=4000 (positives and negatives, 4000 each), and SIZE=20×20. For the evaluation of the remaining parameters, those already evaluated were fixed to the chosen values, while the remaining (except the one under evaluation) were kept as above. For all trainings, HR was fixed to 0.998, and ST was fixed to 15 (except when FA was being evaluated). After 7 training rounds (one for each considered parameter), the parameter set chosen for FIP detection was determined as those in the black nodes of the tree in Fig. 10. Recall and FP shown for each training is relative to the test set.

The choice of parameter values was based not only on recall and false positives but great importance was given also to the simplicity of the resulting classifier. Besides, the number of false positives is far less important than the number of hits in the sense that a post-processing algorithm can deal much better with the former than with the latter. Cascade balance (i.e., uniformity of the number of evaluated features among consecutive stages) was also taken into account given that this desirable characteristic enhances the selective processing focus capability of the cascade approach.

E. Post-processing

The results of the detection phase are supplied to an algorithm that will analyze them and search for groups of 3 matches aligned in an appropriate way to form a QR code symbol. In this step, the size, distance and angles between the matches are analyzed according to the following algorithm:

1) The center of each FIP is considered as a vertex
2) Two vertices are linked if, and only if, their FIPs have similar size (less than 25% width difference) and the distance between them, taking into consideration the FIP size, is compatible with the QR code structure (distance is up to 18 times the width of the FIP[1])
3) For each pair of edges with a common extremity, check whether their angle and length are compatible with the QR code structure. These pairs define a triplet of vertices, candidate to QR code.

A triplet in the QR code candidate set is considered to be a QR code if and only if it appears three times in the candidate set, that is, they correspond to the three vertices of a isosceles right triangle obeying the mentioned criteria.

IV. Experimental Results

In this section we present and discuss the main results obtained with respect to FIP detection (first step) and QR code detection (second step).

A. FIP detection

Training and test of classifiers for FIP detection followed the procedure detailed in Section III-D. Recall and false positives reported here refer to a test set independent of the training set, as described before.

Figure 11 shows the performance (recall and FP) of the twenty classifiers obtained during training parameter assessment described in Section III-D. Arrows with numbers highlight some results. Arrow 1 corresponds to the classifier with best Recall/FP ratio. The small number of FP may be a

[1]The version 40 of QR code is capable of encoding 4296 alphanumeric characters and is composed by 177x177 modules where distance between FIP centers is approximately 168 modules or 18.6 FIP widths.

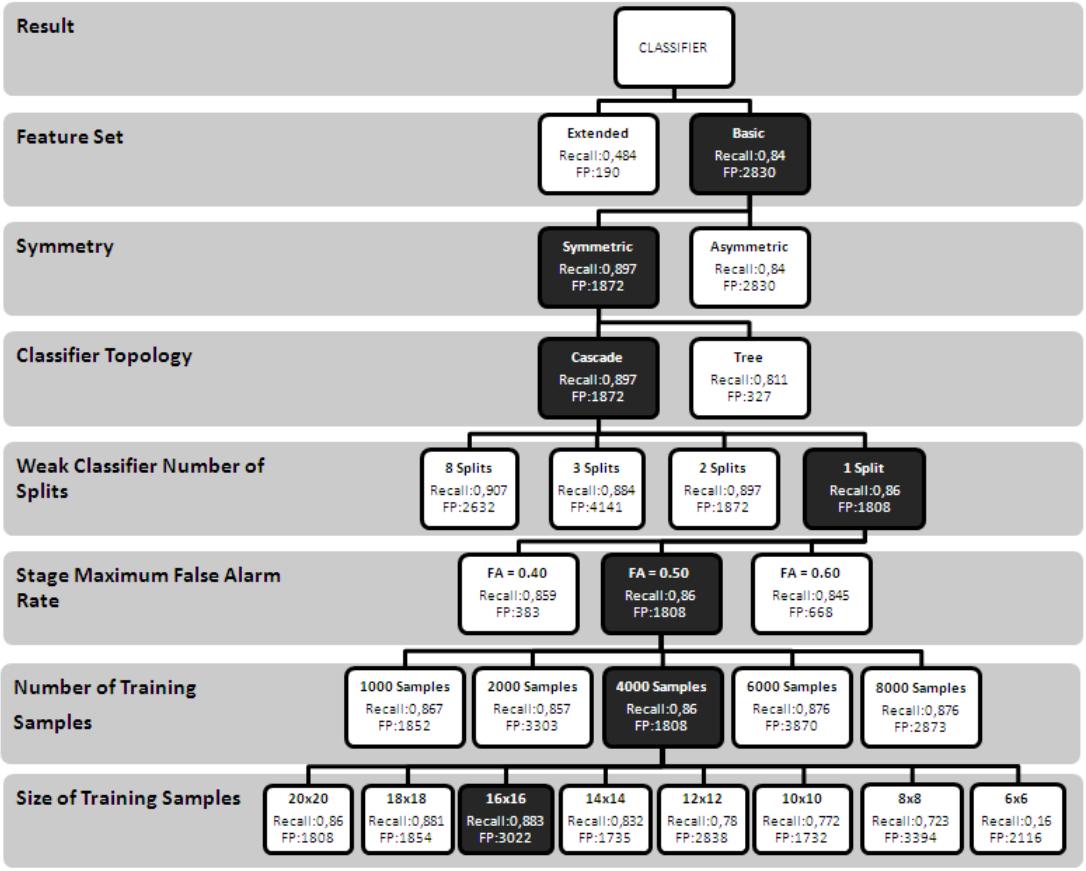


Fig. 10. Parameter choice order: at each level, the chosen value for the respective parameter is in the black node (see text).

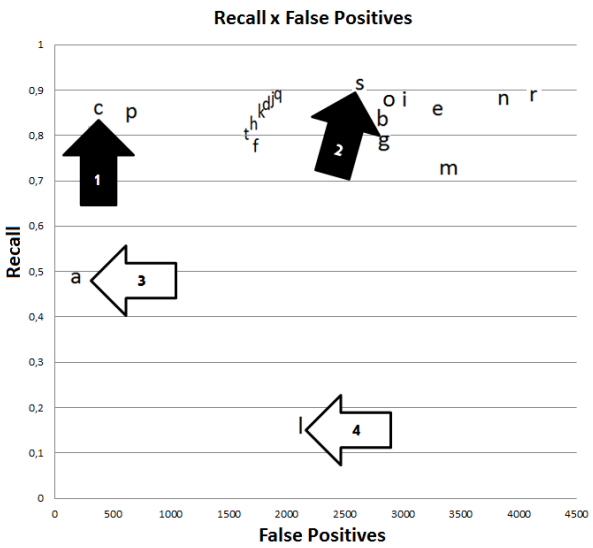
consequence of imposing false alarm limit to 0.4 (FA=0.4) for each stage of the cascade. Arrow 2 points to the best recall obtained, corresponding to a more complex classifier with 8 splits (NS=8). It is noteworthy to mention that a simpler classifier (NS=1) presented difference in recall not superior to 5%. Arrow 3 points to the classifier trained with the extended feature set. Its poor performance on test images clearly indicates that there is strong overfitting to the training set. Arrow 4 points to the classifier trained with minimum sample size equal to 6×6. Its performance indicates that a too small sample size may not be adequate. The conjunction of small sample size and assumption of feature symmetry may have aggravated its performance.

When sample symmetry is supposed, the main features in the cascade classifier are very different from those obtained without the supposition of symmetry. Figure 12 shows the two main features for each case. The average total number of features in the classifier cascade was 110 and, regardless of the parameters used in training, there were no significant changes in this number. The only exceptions were for training samples of size 8×8 and 6×6 that demanded much more features for the specified hit and false alarm rates.

Figure 13 shows the performance of classifiers obtained with training data set consisting of 4000 negative samples and varying number of positive samples. All positive samples were derived from the base set of 285 FIPs, by applying some random variations as described in Section III-C. Training parameters were fixed to the same values of classifier `k` in Fig. 11, except the number of stages (ST) that was set to 17. Recall and false positives are average values of five training runs, with respect to a fixed test set. Although oscillation is present, a clearly increasing trend, with increasing number of positive training samples, can be observed in the recall. The standard deviation in the recall was around 3% for all values considered for N, being slightly larger for $N = 285$. Since each training set was generated randomly, there is no guarantee that a smaller training set is contained in a larger one, and oscillation in the curve may be partly due to this fact. This experiment supports the hypothesis that a relatively small number of FIP samples (in our case, 285) may be sufficient to obtain classifiers with good detection rate.

B. Real image processing

FIP detection rate superior to 90% on test images are achieved with some classifiers. Figures 14 to 16 show examples of resulting images. Squares in the images correspond to FIP candidates detected in the first step while line segments indicate pairwise relations taken into consideration to decide if the corresponding FIPs are or not corners of QR codes. An arrow points to a place in which a QR code was detected.



#	MODE	MTS	SYM	NS	FA	ST	SAMPLES	SIZE
a	ALL	0	N	2	0.5	15	4000	20
b	BASIC	0	N	2	0.5	15	4000	20
c	BASIC	0	Y	1	0.4	13	4000	20
d	BASIC	0	Y	1	0.5	15	1000	20
e	BASIC	0	Y	1	0.5	15	2000	20
f	BASIC	0	Y	1	0.5	15	4000	10
g	BASIC	0	Y	1	0.5	15	4000	12
h	BASIC	0	Y	1	0.5	15	4000	14
i	BASIC	0	Y	1	0.5	15	4000	16
j	BASIC	0	Y	1	0.5	15	4000	18

#	MODE	MTS	SYM	NS	FA	ST	SAMPLES	SIZE
k	BASIC	0	Y	1	0.5	15	4000	20
l	BASIC	0	Y	1	0.5	15	4000	6
m	BASIC	0	Y	1	0.5	15	4000	8
n	BASIC	0	Y	1	0.5	15	6000	20
o	BASIC	0	Y	1	0.5	15	8000	20
p	BASIC	0	Y	1	0.6	23	4000	20
q	BASIC	0	Y	2	0.5	15	4000	20
r	BASIC	0	Y	3	0.5	15	4000	20
s	BASIC	0	Y	8	0.5	15	4000	20
t	BASIC	10	Y	2	0.5	15	4000	20

Fig. 11. Recall × FP for classifiers obtained with different training parameters.

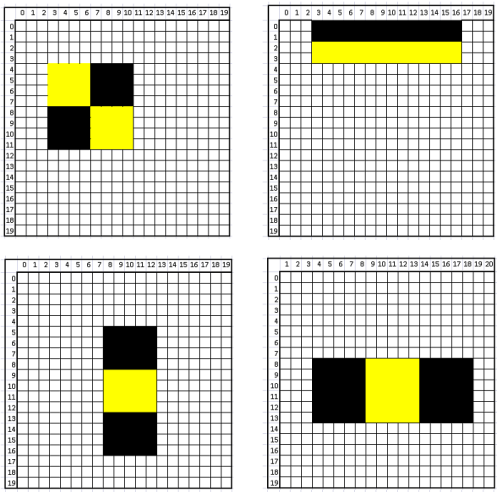


Fig. 12. The two main features when sample symmetry is (top row) and is not (bottom row) supposed.

These results were obtained with a classifier trained with the same parameters of classifier `i` in Fig. 11, except the number of stages (ST) that was set to 17. The choice of this classifier was based on the fact that its performance in recall was very close to the best one observed (classifier `s`) and its simplicity (number of splits equal to 1 against the 8 splits of classifier `s`). Furthermore, classifier `i` considers a smaller window size (16×16), and hence it should be able to recognize smaller

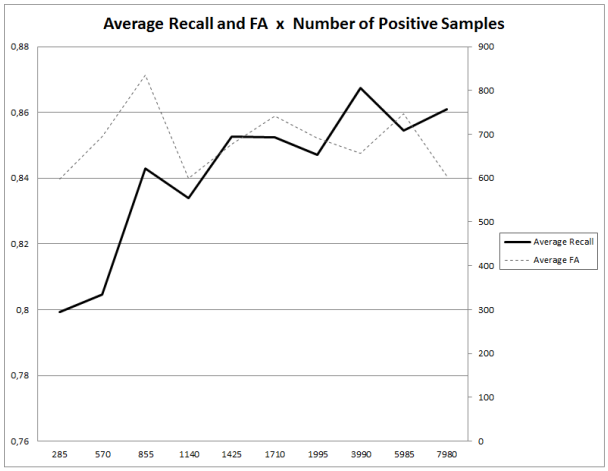


Fig. 13. Effect of augmenting the number of positive samples. Average of five training runs.

FIPs. These images as well as additional resulting images can be found from URL http://www.vision.ime.usp.br/demos.

Fig. 14. FP=5 and QR code correctly detected.

Fig. 15. FP=0 and QR code correctly detected.

1) Video processing and frame rate: Considering that important applications such as aiding visually impaired people depends on real-time processing, we have also evaluated the

Fig. 16. FP=12 and QR code correctly detected.

classifier performance on video frames of size 640×480. Figure 17 shows the average processing time per frame of classifier $\mathtt{m}$ in Fig. 11, when detection sample size is varied from 8×8 to 20×20. Variation of minimum sample size in detection

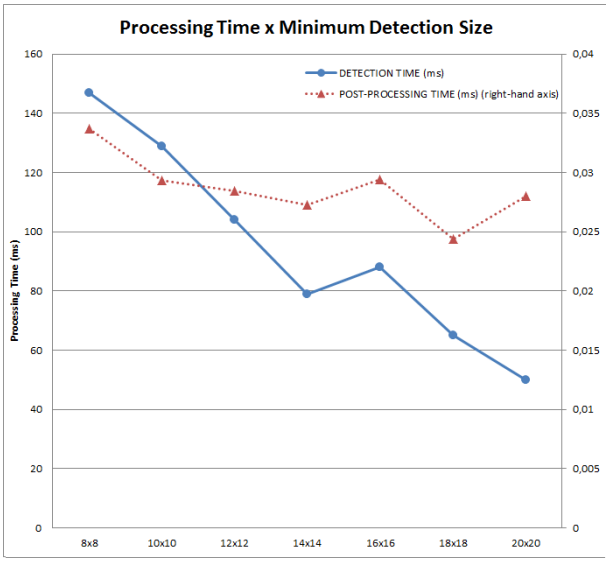


Fig. 17. Processing time for a fixed classifier, with respect to varying sample size in detection.

has strong impact on overall processing time (minimum around 50ms, maximum around 150ms, and average around 125ms per frame). The processing time is inversely proportional to sample size, with the post-processing time (right side axis) corresponding to about 0.003% of the detection time (left side axis). The average number of FP per frame was around 9, explaining the extremely fast post-processing speed.

V. Conclusion

We have investigated the use of Viola-Jones' framework for the detection of QR codes in arbitrarily acquired images. Classifiers obtained with a base training set consisting of only 285 FIP samples exhibit accuracy close or over 90% for the detection of FIPs, with controlled number of false positives. FIPs of size 8×8 as well as very large ones, including rotated ones and those with some perspective distortion, can be robustly detected. A post-processing step that evaluates geometrical constraints among the detected FIPs and decides if a triplet of FIPs are actually corners of a QR code symbol has been proposed and visual inspection indicates promising results. The overall processing time is compatible with real-time application needs.

As future steps of this work, we plan to investigate the effects of training set built from rotated FIP samples, classifier performance variation for different sizes of base training set, a post-processing step that can detect QR codes even when one of the corners is missed in the FIP detection step, and detection of QR codes in video, exploiting temporal coherence.

Acknowledgment

This work is supported by the National Council for Scientific and Technological Development (CNPq), Brazil. L. F. F. Belussi thanks Petrobras for its continuous incentive and support. N. S. T. Hirata is partly supported by the CNPq.

References

[1] Y. Ebrahim, W. Abdelsalam, M. Ahmed, and S.-C. Chau, "Proposing a hybrid tag-camera-based identification and navigation aid for the visually impaired," in *Second IEEE Consumer Communications and Networking Conference (CCNC)*, 2005, pp. 172 – 177.

[2] H. S. Al-Khalifa, "Utilizing QR Code and Mobile Phones for Blinds and Visually Impaired People," in *Proceedings of the 11th International Conference on Computers Helping People with Special Needs*. Springer-Verlag, 2008, pp. 1065–1069.

[3] J. Coughlan, R. Manduchi, and H. Shen, "Cell phone-based wayfinding for the visually impaired," in *1st International Workshop on Mobile Vision, in conjunction with ECCV*, Graz, Austria, 2006.

[4] O. Pârvu and A. G. Bălan, "A method for fast detection and decoding of specific 2D barcodes," in *17th Telecommunications Forum, TELFOR 2009*, 2009, pp. 1137–1140.

[5] X. Liu, D. Doermann, and H. Li, "A camera-based mobile data channel: capacity and analysis," in *Proceeding of the 16th ACM International Conference on Multimedia*, 2008, pp. 359–368.

[6] E. Ohbuchi, H. Hanaizumi, and L. A. Hock, "Barcode readers using the camera device in mobile phones," in *Proceedings of the 2004 International Conference on Cyberworlds*. IEEE Computer Society, 2004, pp. 260–265.

[7] Y. Liu, J. Yang, and M. Liu, "Recognition of QR Code with mobile phones," in *Chinese Control and Decision Conference (CCDC 2008)*, 2008, pp. 203 –206.

[8] C.-H. Chu, D.-N. Yang, and M.-S. Chen, "Image stablization for 2D barcode in handheld devices," in *Proceedings of the 15th International Conference on Multimedia*, 2007, pp. 697–706.

[9] Y. Xue, G. Tian, R. Li, and H. Jiang, "A new object search and recognition method based on artificial object mark in complex indoor environment," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, july 2010, pp. 6648 –6653.

[10] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518.

[11] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Intl. J. Computer Vision*, vol. 38, no. 1, p. 1533, 2000.

[12] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *IEEE ICIP*, 2002, pp. 900–903.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.

[14] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," Microprocessor Research Lab, Intel Labs, Intel Corporation, MRL Technical Report, 2002.

[15] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.