QR Code Detection in Arbitrarily Acquired Images

Luiz F. F. Belussi, 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*—Applications of Quick Response (QR) codes enable rich context interaction through creation of links between physical objects and Internet resources. In spite of the widespread use of this kind of barcode, applications for visually impaired people and robots are not common because existing decoders assume that the symbol is properly framed during image acquisition. This work proposes a two-stage component-based approach to perform accurate detection of QR code symbols in arbitrarily acquired images. In the first stage a cascade classifier to detect parts of the symbol is trained using the rapid object detection framework proposed by Viola-Jones. In the second stage, detected patterns are aggregated in order to evaluate if they are spatially arranged in a way that is geometrically consistent with the components of a QR code symbol. An extensive study of parameter variation of both stages was performed and the results were analyzed in terms of precision, recall and computational efficiency. The proposed QR code detector achieved average recall of 91.7% with precision of 76.8% while being capable of processing a 640×480 pixels video stream at 22 fps. These results support implementation of real-time applications that assist visually impaired people and robots in mobile hardware, allowing them to have access to the wealth of information available through QR codes in multiple medium.**

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

I. Introduction

Quick Response (QR) code is a type of two-dimensional code introduced by Denso Wave in 1994 [1]. They were initially used by the automotive industry to track vehicle parts during the manufacturing process. However nowadays they are most commonly used to encode hyperlinks, notably in the advertising industry, to connect places and objects to websites containing additional context relevant information. A QR code symbol, according to the ISO/IEC standard 18004, has a general structure that comprises data, version information and error correction code words. Their internal organization with respect to structural components may vary depending on the amount of data encoded. The three square patterns at the corners of the symbols are called finder patterns (FIPs) and are meant to be easily found under bad imaging conditions. Figure 1 shows examples that illustrate distinct QR code versions and the number of encoded characters.

The existing decoders, easily found for mobile devices, are able to work correctly only if codes are properly framed, with code region corresponding to at least 30% of the image. To make this kind of technology useful for visually impaired people or autonomous navigation of robots, detecting the presence of a code in an image is a necessary step prior to the adequate framing process.

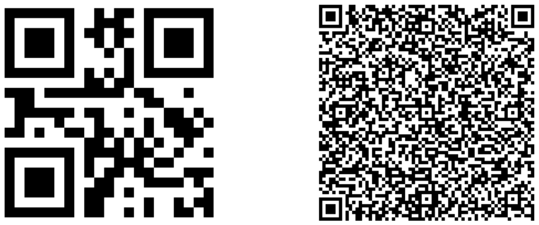

Fig. 1. Different versions of QR code symbols: with 10 (left) and with 500 (right) encoded alphanumeric characters.

In related literature, the majority of work that mention QR code recognition or detection is actually concerned with improving image quality or determining the exact contours of the code rather than finding (i.e, deciding whether there is or there is not) a QR code symbol in an image [2], [3], [4]. In fact, most of the images considered in those works are images acquired with the specific intent of capturing only the code symbol. Some few works that deal with the problem of finding QR codes propose solutions that rely on auxiliary information such as visual cues or RFID tags [5], [6].

This work[1] addresses the problem of detecting QR codes in arbitrarily acquired images without relying on auxiliary cues. The proposed approach is a two-stage component-based detection method. In the first stage, candidate FIPs are detected and in the second stage information about detected FIPs are integrated in order to decide whether subsets of three of them correspond to corners of a QR code or not.

In the first stage it is desirable to maximize true positives (TP) while maintaining a controlled number of false positives (FP). The rapid object detection framework proposed by Viola and Jones [9] is particularly interesting for this stage because it allows controlling TP and FP, besides being extremely fast and also invariant to scale and to illumination variations.

One of the contributions of this work, described in Sec-

[1] This work was developed by the first author during his Master studies. His dissertation, in Portuguese, is available at http://www.teses.usp.br/teses/disponiveis/45/45134/tde-28112012-180744. The authors have also published two papers, [7] and [8], during the development of this work. Sample images of QR code detection using the proposed method, not included in this paper due to space limitations, are available at http://www.vision.ime.usp.br/demos.

tion II, is an empirical evaluation of several training parameters of the Viola-Jones' framework for the detection of FIPs. Another contribution is a graph-based formalization for the problem of aggregating the detected FIP candidates in order to decide if they correspond to parts of QR codes. The proposed formalization, described in Section III, relies on the analysis of geometrical relationships among detected candidate FIPs to verify if there are subgroups of three of them spatially arranged as three corners of a square region. Experimental results of the proposed approach are presented in Section IV. QR code detection rate superior to 90% was observed on test images using a reduced set of training samples, suggesting that even better rates can be achieved in the future. Conclusions are presented in Section V.

II. Detection of Finder Patterns (FIPs)

Some algorithms for recognition of QR codes rely on the regularity of alternation between black and white pixels that are characteristic to FIPs in order to find them and then determine the exact contours of the code. However, that regularity may be disrupted in low quality images. Similarly, extraction of features such as histograms or frequency information from the code area can be greatly affected as well.

In this work we propose the use of Viola-Jones' rapid object detection framework [9] for detecting FIPs. FIPs are typically the largest structures within a code and have a fixed shape. Viola-Jones' approach is appropriate because it meets some requirements for real-time QR code detection such as speed, invariance to scale and illumination variations.

The obtained classifiers are based on a set of simple Haar-like features constrained within a fixed size window. Integral images are used for the fast computation of the feature values, requiring only summations and no multiplications, and a cascaded approach for training a classifier is used (each stage is trained in such a way as to achieve fixed TP and false alarm (FA) rates, using a boosting algorithm). Application of the classifiers is also very fast in practice due to the cascaded structure of the classifier that discards the majority of negative samples in its early stages, eliminating the need for calculating responses of all stages to every sample considered. Scale invariance is achieved by considering different window sizes at each position in the image and processing samples after reducing all of them to the fixed size considered by the classifier [9].

Viola-Jones framework is widely known for face detection and the literature concerning the training of the cascade classifier often recommends training practices, parameters and sets, suitable for that application domain [10]. Due to the large number of parameters in the training process, an exhaustive assessment of possible combinations of parameter values is not feasible. For the training of FIP detectors, we have used an approach similar to the one used for face detection in [11]. That is, a sequential order as well as default values for the parameters to be evaluated were initially fixed. Then, the effect of variations in each of the parameters, one at a time, were assessed using a validation set. Once an optimal value was estimated for a parameter, its value was fixed, and evaluation proceeded to the next parameter, till the last one. All experiments for FIP detection were performed using the implementation of Viola and Jones method available in OpenCV. Additional details can be found in the dissertation. The chosen values are listed below:

- Feature Set: Basic
- Symmetry: Symmetric
- Classifier Topology: Cascade
- Weak Classifier Number of Splits: 1
- Stage maximum False Alarm Rate: 0.50
- Number of Training Samples: 4000
- Size of Training Samples: 16×16.

III. Component aggregation method

Analysis of the arrangement of FIP candidates detected in the first stage is performed in the second stage in order to determine if the candidate FIPs correspond to the corners of a QR code.

Each FIP candidate is represented here as a triple (T, x, y) where T is the size of a square region, and x and y are the coordinates of the center of that square in the image.

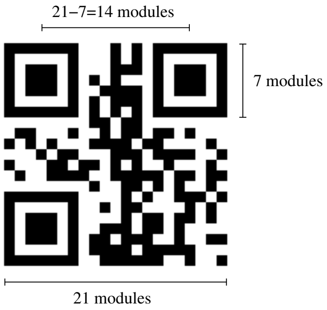

Fig. 2. QR code version 1 is composed of 21×21 modules.

In a noise and deformation free image, two FIPs of a same QR code have the same size and the maximum distance between them is limited by a factor related to FIP size. A FIP corresponds to 9×9 modules, considering the white region around it. The smallest version of QR code (version 1) can accommodate 21 modules (see Fig. 2) while the largest one (version 40) can accommodate 177 modules in its width. Thus, the minimum number of modules between the center of two adjacent FIPs is 21-7=14, and the maximum is 177-7=170. Hence, the distance between two FIPs in adjacent corners is at least $(T/9)14 \approx 1.6T$ and at most $(T/9)170 \approx 19T$. Since they may be on opposite corners, the maximum distance is therefore $19\sqrt{2}T$.

Moreover, two edges linking the three FIPs of a QR code are either orthogonal each other (two sides of the square region) or they form an angle of $45°$ (one of the sides and the diagonal of the square region). Given two FIPs of a QR code, there are six possible positions for the third FIP, as shown in Fig. 3.

Let $F_1 = (T_1, x_1, y_1)$ and $F_2 = (T_2, x_2, y_2)$ be two FIP candidates. Let $\overline{F_1 F_2}$ denote the line segment that links (x_1, y_1) and (x_2, y_2), and $|\overline{F_1 F_2}|$ denote its length. Note that

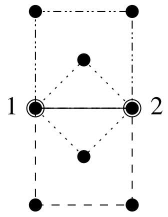

Fig. 3. Given two FIPs (indicated as 1 and 2) of a QR code, the six possible positions for the third FIP are represented by dark circles.

$\overline{F_2\,F_1} = \overline{F_1\,F_2}$. We say that F_1 and F_2 satisfy the **size criterion** if and only if $T_1 = T_2$. If $T_1 = T_2 = T$ we say that they satisfy the **distance criterion** if and only if $1.6\,T \leq dist\big((x_1, y_1), (x_2, y_2)\big) \leq 19\sqrt{2}\,T$. Given a third FIP candidate, F_3, let $\overline{F_1\,F_2}$ and $\overline{F_1\,F_3}$ be the two line segments with common extremity (x_1, y_1), θ be the angle between them, and $P = \dfrac{min\{|\overline{F_1\,F_2}|, |\overline{F_1\,F_3}|\}}{max\{|\overline{F_1\,F_2}|, |\overline{F_1\,F_3}|\}}$. Then, we say that $\{\overline{F_1\,F_2}, \overline{F_1\,F_3}\}$ satisfies the **orientation criterion** if and only if
(i) $\theta = 90^\circ$ and $P = 1$ (i.e., $|\overline{F_1\,F_2}| = |\overline{F_1\,F_3}|$), or
(ii) $\theta = 45^\circ$ and $P = \frac{\sqrt{2}}{2}$.

Given that three candidate FIPs are actually corners of a QR code, then pairwise they must satisfy the size and distance criteria and the edges of the triangle also satisfy, pairwise, the orientation criterion. Therefore, finding QR code candidates can be cast as a problem of finding subsets of three FIPs, satisfying the three necessary conditions above.

The formulation proposed for solving this problem is based on graphs. FIP candidates are represented as vertices and two vertices are linked if the pair of respective FIPs satisfy the size and the distance criteria. Note, however, that not all triplet of candidate FIPs satisfying pairwise the size and distance criteria are corners of a QR code. Therefore, the solution of the problem does not correspond to all cliques of size three in the graph. There is a need to, additionally, verify whether the edges satisfy or not the orientation criterion.

Proposition 1. *Let $\{F_1, F_2, F_3\}$ be a set of FIP candidates that pairwise satisfy the size and distance criteria. Then, $\{\overline{F_1\,F_2}, \overline{F_1\,F_3}\}$ satisfies the orientation criterion if and only if $\{\overline{F_2\,F_1}, \overline{F_2\,F_3}\}$ and $\{\overline{F_3\,F_1}, \overline{F_3\,F_2}\}$ also satisfy the orientation criterion.*

Proof: : If $\{\overline{F_1\,F_2}, \overline{F_1\,F_3}\}$ satisfies the orientation criterion, then either they are two sides of a square or they are one side and the diagonal of a square. If they are two sides of a square, then the third edge $\overline{F_2\,F_3}$ is the diagonal and obviously it satisfies condition (ii) of the orientation criterion with $\overline{F_1\,F_2}$ and also with $\overline{F_1\,F_3}$. If they are one side and the diagonal of a square, obviously the third one is another side of the square that satisfies condition (i) of the orientation criterion with the edge that is a side of the square, and condition (ii) of the orientation criterion with the edge that is the diagonal of the square. ■

An important consequence of this proposition is that given that three FIP candidates satisfy pairwise the size and distance criteria, just one pair of edges satisfying the orientation criterion is sufficient to determine that the three FIPs correspond to a QR code candidate.

Moreover, in an ideal condition, given that a FIP is part of a QR code, it can not be part of another QR code. Thus, once a triplet satisfying the code condition is found, they could be removed from the graph together with all edges adjacent to it. However, there may be situations such as the one shown in Fig. 4, in which the first detected triplet is a false positive. In order to avoid loosing the other, an edge should be removed only after all edges adjacent to it are analyzed.

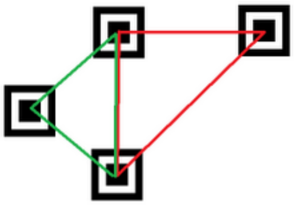

Fig. 4. An edge that may be part of two subsets of three FIP candidates satisfying the conditions.

In practice, the size and distance values obtained from an image are not precise due to, for instance, imaging conditions, image resolution, rotation and perspective distortions. Therefore, the criteria should allow some tolerance to accommodate imprecision due to these conditions.

- (size) Let $T = \max\{T_1, T_2\}$ and $t = \min\{T_1, T_2\}$. The size difference between two FIP candidates is allowed to vary up to a given percentage, i.e., $T - t \leq \varepsilon T$. For instance, $\varepsilon = 0.2$ means that maximum allowed size difference is at most 20%.
- (distance) If the maximum code version to be detected corresponds to the largest version, then distance should satisfy $1.6\,T \leq dist\big((x_1, y_1), (x_2, y_2)\big) \leq 19\sqrt{2}\,T$, where $T = \max\{T_1, T_2\}$.
- (orientation): a small variation in the angle as well as in the length of the edges should be allowed. These can be parametrized by one single parameter d, by rewriting the orientation condition as (i) $|90^\circ - \theta| \leq d$ and $1 - \dfrac{cos(45^\circ - d) - cos(45^\circ + d)}{cos(45^\circ)} \leq P \leq 1$ or (ii) $|45^\circ - \theta| \leq d$ and $cos(45^\circ + d) \leq P \leq cos(45^\circ - d)$.

A. Aggregation algorithm

The proposed solution for aggregating FIP candidates is presented in Algorithm 1. Lines 1 to 3 correspond to vertex set creation. Lines 4 to 13 correspond to the creation of edges, whenever a pair of vertices satisfy the size and the distance criteria, with respective tolerances. The loop structure from line 14 to 23 is the main part of the algorithm, where cliques of size three whose edges pairwise satisfy the orientation criterion are found. For each edge (u, v), all edges (u, v') that are adjacent to one of its extremities (u) are verified. For each of such pairs of edges (i.e, $\{(u, v), (u, v')\}$), first the existence of the third edge (i.e, edge (v, v')) is verified. If the third edge is not there, that means that either the two vertices, v and v' do not satisfy the size or distance criteria, or that the third edge has already been processed (in this case, the triplet $\{u, v, v'\}$ is already in the output list). If the third edge (v, v') is present, and if the first two satisfy the orientation condition, then the subset of the three vertices $\{u, v, v'\}$ is added to the output list. Note that there is no need to examine edges that are adjacent to the other extremity (v) since those will be evaluated later. The algorithm ends when all edges are processed.

Algorithm 1: FIP candidate aggregation algorithm.

```
Input: a list of FIP candidates, each represented by a
       triple $F_i = (T_i, x_i, y_i)$, and tolerance parameters $\varepsilon$
       and $d$
Output: a list of subsets of the form $\{F_i, F_j, F_k\}$
        satisfying the size, distance, and orientation
        criteria, with respective allowed tolerances.

foreach $F_i$ do
    create a vertex $v_i$ and mark it as unprocessed
end

while Unprocessed vertices $u$ remaining do
    foreach unprocessed vertex $v$, $v \neq u$, do
        if $\{u, v\}$ satisfies the size criterion with tolerance
        $\varepsilon$ then
            if $\{u, v\}$ satisfies the distance criterion then
                create edge $(u, v)$;
            end
        end
    end
    mark vertex $u$ as processed;
end

while edges $(u, v)$ remaining do
    foreach edge $(u, v')$, $v' \neq v$, do
        if there exists edge $(v, v')$ then
            if edges $(u, v)$ and $(u, v')$ satisfy the
            orientation criterion with tolerance $d$ then
                add subset $\{u, v, v'\}$ to the output list
            end
        end
    end
    Remove edge $(u, v)$;
end
```

IV. Experimental Results

This section presents the experiments carried out in order to determine suitable parameter values for both the FIP detection stage and aggregation stage, and to evaluate the final QR code detector. The final QR code detector was evaluated in terms of precision and recall in a cross-validation scheme. Its computational performance for processing a series of video streams has also been evaluated.

A. FIP detection parameters

The FIP detector obtained as described in Section II allows one to specify a scaling factor (SF) and the number of minimum overlapping detections (ND). The scaling factor refers to how much sample size is decreased in each processing iteration (for instance, $SF = 1.10$ indicates that at each location samples size vary from the largest possible to the minimum detection size, being 10% smaller from one step to other). As multiple positive responses may occur around a single instance of the pattern in the image, ND specifies the minimum number of detections in order to that region be considered a single detection.

A FIP detector was trained with the chosen parameter values (see Section II) using a set of 462 positive samples (cropped FIP regions) and 4000 negative samples from 820 background images with no QR codes (for each stage, the training algorithm selects negative samples from these background images).

To determine suitable values for parameters SF and ND, the trained FIP detector was evaluated on a independent set consisting of 74 images containing 222 positive samples (exactly one complete QR code in each image).

a) Scaling factor (SF): Figure 5 shows that the scaling factor (SF) directly impacts the quality of results. A wide range of result quality was observed from SF=1.05 (recall: 0.98 / FP: 628) to SF=1.30 (recall: 0.86 / FP: 93). As the scaling factor is incremented, both recall and FP decrease. Considering that high recall and small FP are desirable, scaling factor of 1.10 as highlighted in the chart (recall: 0.99 / FP: 290) was adopted for the successive experiments.

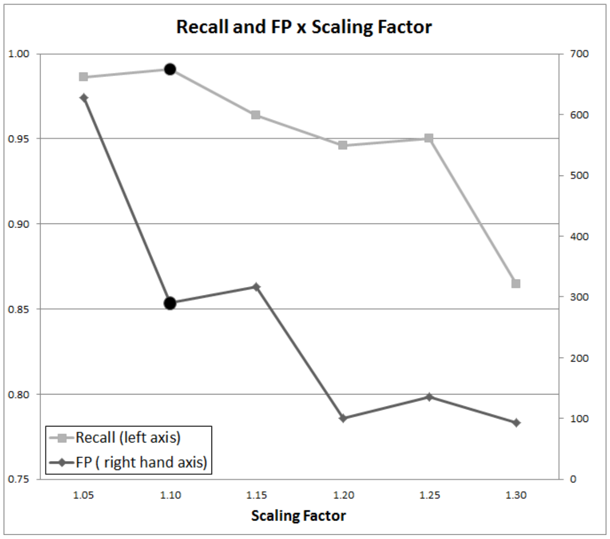

Fig. 5. Recall and FP of the trained FIP detector varying the scaling factor used during FIP detection.

b) Overlapping detections (ND): Figure 6 shows a ROC curve comparing recall and FP while ND varies from 1 to 200. A possible choice for ND would be 53 as highlighted in point A. However, it should be noted that these recall and FP results are relative to to FIP detection. They will be processed in the second stage by the FIP aggregation algorithm and therefore it is much more important to have a high recall than a low FP in the first stage. Based on that reasoning, ND=30 (highlighted in point B of Fig. 6) was chosen.

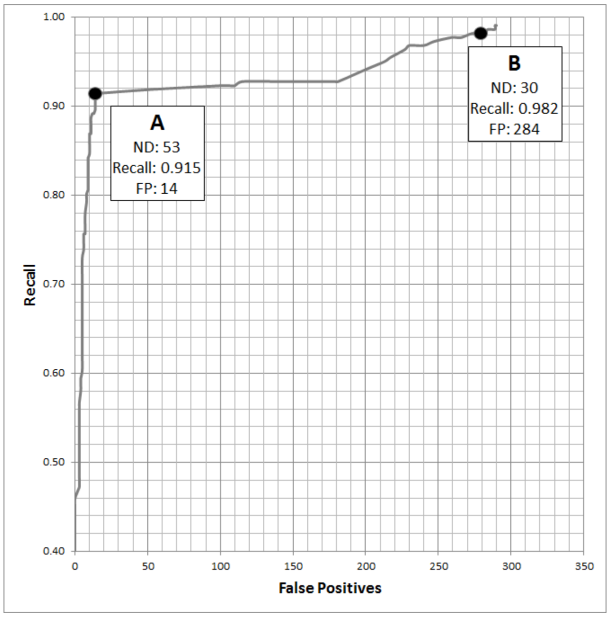

Fig. 6. Recall × FP for the trained FIP detector varying ND from 1 to 200.

B. QR code detection parameters

Having FIP detector parameters fixed as determined in the previous section, an exclusive validation set composed of 45 images with exactly one complete QR code each was used to perform parameter tuning for complete QR code detection. Individual variations in the tolerances ε and d respectively for the size and orientation criteria were evaluated. The best performing values for these parameters were chosen as the final parameter values for FIP aggregation: $d = 0.25$ and $\varepsilon = 0.40$ and $ND = 30$, with FIP detection parameters $SF = 1.1$ and $ND = 30$.

C. Final Performance of the QR Code detector

The performance of Viola-Jones cascade classifier may vary considerably depending on the quality of the samples used during training. In order to make a solid evaluation of the performance of the proposed QR code detector, a 5-fold cross-validation was performed. A set of 385 images of varying quality containing one QR code each and a total of 1155 FIPs was split in the ratio of 4:1 in five distinct ways so that $4/5$ of the samples were used to train the FIP detector and $1/5$ were used to evaluate the whole QR code detector (using the parameter values SF=1.1, d = 0.25, ε=0.40, and ND=30, described in Section IV-B). Table I shows the results of the cross-validation.

An example of correct detection is shown in figure 7.

D. Video data processing

In the proposed method, the total processing time is dominated by the FIP detection stage. FIP detection is greatly influenced by scaling factor and the minimum size of FIP to be detected.

TABLE I
Cross-validation for complete QR code detection

Round	TP	FN	FP	Recall	Precision
Round 1	68	9	15	0,8831169	0,819277108
Round 2	69	8	17	0,8961039	0,802325581
Round 3	73	4	8	0,9480519	0,901234568
Round 4	71	6	21	0,9220779	0,77173913
Round 5	72	5	61	0,9350649	0,541353383
Average	70,6	6,4	24,4	0,9168831	0,767185954

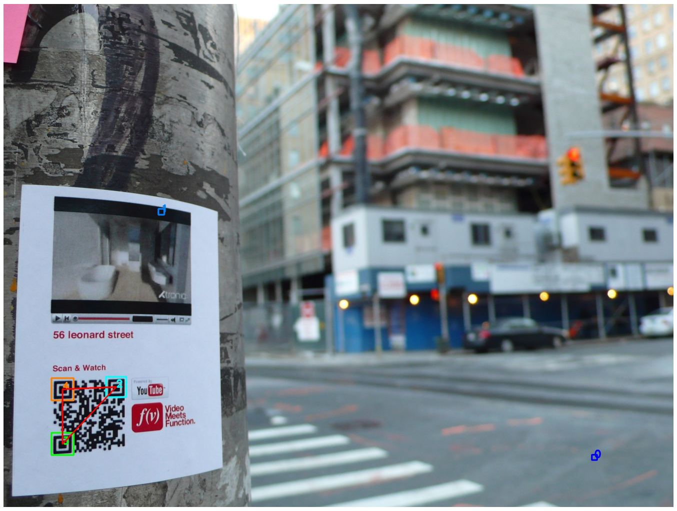

Fig. 7. Example of correct detection of a QR code, with two false positive FIPs.

The results in this section quantify the influence of these parameters in computational performance showing that these parameters can be used to adjust the trade-off between result quality and the frame rate of processed video, a desirable characteristic when a great variety of devices should be considered for potential applications.

The experiments were run in a virtual machine inside a dual-core computer and the results are the average values for 1000 frames. The video, captured by a standard webcam, contains a scene in which two QR code symbols are moved around in various ways. The frame size is 640 × 480.

1) Minimum detection size in video processing: Figure 8 shows the processing time variation for different values of minimum detection size, fixing scaling factor to 1.20. The average processing time per frame varies from 22 ms to 103 ms and the time spent on the second stage (right hand axis) is only approximately 0.8% of the processing time of the FIP detection stage.

2) Scaling factor in video processing: The scaling factor was varied from 1.05 to 1.30 with minimum FIP detection size fixed to 16 × 16 pixels. Figure 9 shows that processing time ranges from 45 ms to 255 ms per frame reaching frame rates between 3.9 fps and 22.0 fps.

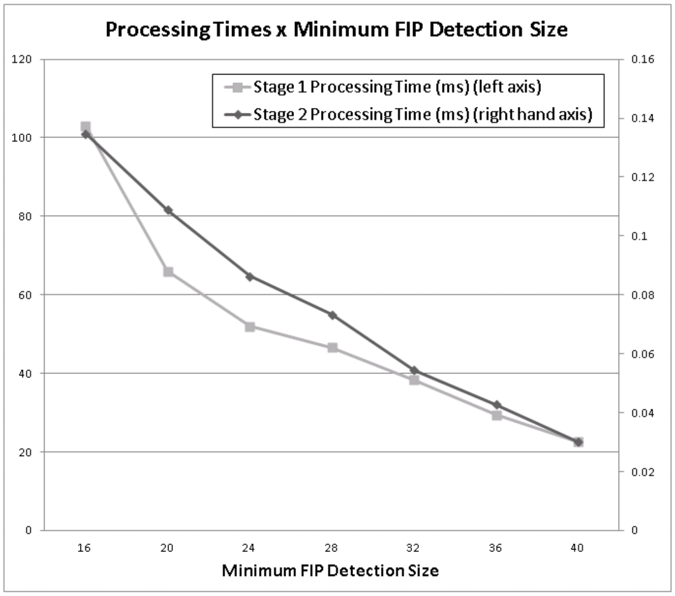

Fig. 8. Processing times of stages 1 and 2 (results in ms, video stream of 640 × 480 pixels, average of 1000 frames).

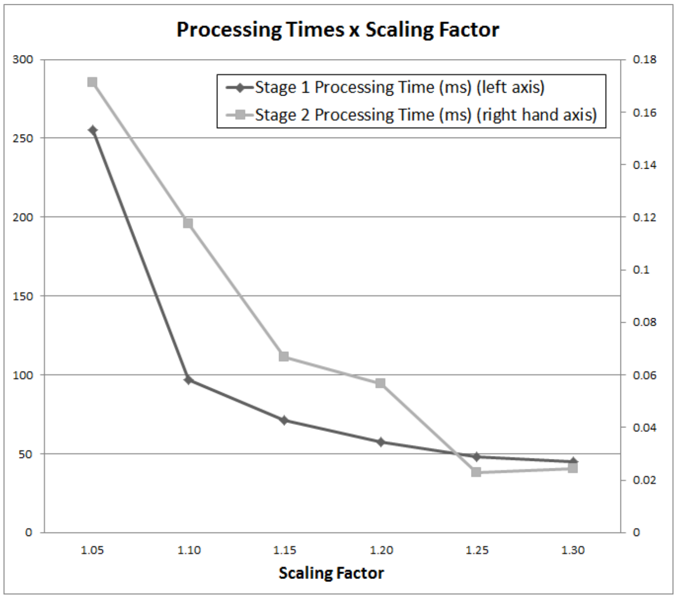

Fig. 9. Processing times of stages 1 and 2 (results in ms, video stream of 640 × 480 pixels, average of 1000 frames).

V. Conclusions

Detecting QR codes in arbitrarily acquired images is a necessary step for important applications that can not assume sighted users. To the best of knowledge, this problem has not been addressed before without relying on auxiliary cues to help detection.

The proposed approach was validated and tested on still images, resulting in QR code detection rates superior to 90% on test images. Part of these undetected codes are consequence of undetected FIPs (by construction, the aggregation algorithm can not detected a QR code if one of its FIPs is not detected). Another part is due to geometrical distortions of the QR code beyond that allowed in the FIP aggregation algorithm. Tests on video stream data show that processing time compatible with acceptable frame rates are obtained.

Since these results were achieved using a FIP detector trained with a relatively small number of samples, there is an expectation that even better FIP detection rates can be achieved. Current results show robustness with respect to scale and illumination variation.

Based on these results and foreseen further improvements, it can be concluded that the proposed approach has great potential to enable several important applications that depend on QR code detection in arbitrarily acquired images.

Acknowledgments

N. S. T. Hirata is partly supported by CNPq.

References

[1] DensoWave, "QRcode.com," http://www.denso-wave.com/qrcode/index-e.html, access on November 17, 2011.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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 *8th World Congress on Intelligent Control and Automation (WCICA)*, july 2010, pp. 6648–6653.

[7] L. Belussi and N. S. T. Hirata, "Fast qr code detection in arbitrarily acquired images," in *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*, 2011, pp. 281–288.

[8] L. Belussi and N. Hirata, "Fast component-based qr code detection in arbitrarily acquired images," *Journal of Mathematical Imaging and Vision*, vol. 45, no. 3, pp. 277–292, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10851-012-0355-x

[9] 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.

[10] S. C. Brubaker, M. D. Mullin, and J. M. Rehg, "Towards optimal training of cascaded detectors," in *9th European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, vol. 3951. Springer, 2006, pp. 325–337.

[11] 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.