Integral-map Based Fast and Robust Screening Method for Endoscope Images

Jia Gu, Lei Wang
Key Lab for Biomedical Informatics and Health Engineering, Shenzhen Institute of Advanced Technology, China, 518067
Jia.gu@sub.siat.ac.cn, wang.lei@siat.ac.cn

Abstract. Texture-based analysis and classification is of vital importance to the early detection of colorectal cancer precursors: flat lesion. However, the traditional way of its computation is very time consuming, which limits the application of using automated CAD (computer aided detection) algorithms in the clinical screening practices, whose one of the major concerns is high detection speed. Therefore in this paper, we present a fast yet robust algorithm for texture classification using discriminative learning models. By adopting the idea of integral-map, the feature extraction and description stages can be dramatically accelerated, thus efficient detection and screening of flat lesions in endoscope images can be ensured. Experimental results show the effectiveness and stability of our approach.

Keywords: Texture classification; Fast computation; Integral map; Endoscope image

1 Introduction

Colorectal cancer is the second leading cause of cancer deaths, ranks third for new cancer cases and cancer mortality for both men and women, and its death rate can be dramatically reduced by appropriate treatment when early detection is available [1]. Among all the precursors of colorectal cancer, flat lesion is especially hard to detect due to lack of protruding shape [2~4], therefore conventional CAD (computer aided detection) system [5, 6] using VC (virtual colonoscopy constructed by CT volume) may not be sufficient for clinical use, since the flat lesions usually is not visible under CT scanning.

We are developing fully automated Computer aided diagnosis systems that mimic doctor’s diagnostic process to assess the severity of abnormalities, using optical endoscope images [7, 8]. One of the major modules is the flat lesion detection and identification, based on texture classification of 2D images. Texture classification is a fundamental building block of image analysis that is frequently applied in a variety of important computer vision applications such as target recognition and robotic vision, and has been extensively studied over the past decades, with many research activities has been published [10~14]. As it is well known, the computational time of texture based image classification, if using all the above mentioned method [10~15], requires
quite excessive time. To overcome this shortcoming, only a few algorithms have been developed to speed up texture classification [16~18]. Dong-Gyu et al. [16] proposed a DCT-based texture description method. Song et al. [17] proposed a fast texture segmentation using genetic programming. However none of the existing research has achieved the level of near real-time. And to the best of the author’s knowledge, not much research has been focused on the clinical application.

Recently, Viola et al. [19] published an exciting research progress on real-time face detection. In their experiment, integral-map and Ada-boosting have been used together to accelerate face detection, and they have achieved the detection rate of 15 frame per second, which is already near real-time. Also, in year 2006, Bay et al [20] proposed a fast algorithm for feature extraction and matching which is invariant to rotation and scaling, utilizing Hessian matrix and integral map. Both algorithms mentioned above use fast calculation based on integral-map. Inspired by their algorithm, we present in this paper a novel approach for fast texture classification, which is a totally different application, but employing similar principle. The main contributions in this paper are twofold. The most significant one is a novel feature extraction and description method using integral map, which permits us to perform texture classification in a near real-time mode. Our second contribution is a seamless integration between doctor’s clinical guidance and software platform, by taking advantage of a completed set of annotations provided by world-class colonoscopist. These pixel-level accurate annotations not only serve as training data for building supervised classification algorithms, but are also used for evaluation and validation.

2 Methods

The organization of this paper is as follows: Section 2.1 and 2.2 briefly recall the related work. In Section 2.3, we illustrate the design and implementation of the novel texture classification method in detail. In section 2.4, we introduce the database that was used in this paper, and how we are making use of doctor’s annotation. We then evaluate the algorithm in section 3 and conclude in section 4.

2.1 Existing method: Integral-map

In Ref. [19], Viola et al employs a fast rectangle feature computation method using an intermediate representation for the image which is later called integral-map [21]. The integral-map at location $x, y$ contains the sum of the pixels above and to the left of $x, y$, inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

(1)

where $ii(x, y)$ is the integral-map and $i(x, y)$ is the original image (see Figure 1a). Using the following pair of recurrences:

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

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

(2)
Figure 1a: The value of the integral-map \([21]\) at point \((x, y)\) is the sum of all the pixels above and to the left. Figure 1b: The sum of the pixels within rectangle \(D\) can be computed with four array references. The value of the integral-map at location 1 is the sum of the pixels in rectangle \(A\). The value at location 2 is \(A+B\), at location 3 is \(A+C\), and at location 4 is \(A+B+C+D\). The sum within \(D\) can be computed as \((4+1-3-2)\)

Using the integral-map any rectangular sum can be computed in four array references (see Figure 1b). Clearly the difference between two rectangular sums can be computed in eight references. Since the two rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features. By using integral-map technique, they have accelerated their face detection system into 15 frames per second.

2.2. Existing method: Hessian features

In Ref [20], Bay et al introduced a fast feature matching method using integral-map. They reform the Hessian matrix as follows:

\[
H(x, \sigma) = \begin{bmatrix}
L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\
L_{yx}(x, \sigma) & L_{yy}(x, \sigma)
\end{bmatrix}
\]

(3)

Where \(L_{xx}(x, \sigma)\) is the convolution of the Gaussian second order derivative \(\frac{\partial^2}{\partial x^2} g(\sigma)\) with the image \(I\) in position \(x\) at scale \(\sigma\), and similarly for \(L_{yy}(x, \sigma)\) and \(L_{xy}(x, \sigma)\). As Simard et al. pointed out in [22], this type of convolution can be significantly accelerated using the following formula:

\[
f * g = \iint (f' * g') = (f''') * \iint g
\]

(4)

Where \(f'\) and \(f''\) are first and second derivatives. Since the integral-map is already the double integral of the image (first along rows and then along columns). The second derivative (first in row and then in column) of the convolution template if it is a box image (with value 1 within the rectangle of interest and 0 outside) yields four delta functions at the corners of the rectangle, which only requires simple addition, and thus can be calculated at very high speed.
Figure 2: Approximation of Gaussian derivatives using box filters. Fig. 2a – Fig. 2c illustrate the theoretical values of Gaussian second derivatives: $L_{xx}(x,\sigma)$, $L_{xy}(x,\sigma)$, $L_{yy}(x,\sigma)$. Fig. 2d – Fig. 2f show the corresponding approximation using box filter.

Following Eq. (4), Bay et al approximate the above mentioned $L_{xx}(x,\sigma)$, $L_{xy}(x,\sigma)$, $L_{yy}(x,\sigma)$ with box filters (see Figure 2 for example). In order to deal with heavy computation, they repeatedly smooth the image with a Gaussian and subsequently sub-sampled to create a list of image pyramids. Due to the use of box filters and integral images, they do not have to iteratively apply the same filter to the output of a previously filtered layer, but instead can apply such filters of any size at exactly the same speed directly on the original image. Therefore, the scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size, which dramatically accelerates the feature extraction and matching algorithm.

2.3 Proposed texture classification method

Inspired by the above mentioned efforts, we now describe our approach for fast texture classification, which include three steps: feature extraction, supervised training and online testing.

(1) Feature extraction based on Zernike moments

Zernike moments (ZMs) have been successfully used in pattern recognition and image analysis due to their good properties of rotation invariance and orthogonality [25]. The rotation invariance property enables the successful feature correspondence under various view-points of image acquisition, while the orthogonality property ensures no redundancy exists among the feature set.

The $nm$th Zernike moments for image $f(x,y)$ are calculated by:

$$Z_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x,y) \overline{V_{nm}^*(\rho,\theta)} dxdy$$

(5)

With $V_{nm}^*(\rho,\theta) = R_{nm}(\rho)e^{-j\theta}$ and
Several fast computation algorithms of Zernike moments have been thoroughly investigated including recently proposed [24]. However in this application, Zernike moments are approximated by box filters [20] in order to take advantage of the efficiency of integral map as much as possible (see figure 3 for illustration of the approximate process, note that we only utilize up to 4th order moments due to computational simplicity).

\[ R_{nm}(r) = \sum_{s=0}^{n-m} \frac{(-1)^s (n-s)!}{s! \left( \frac{1}{2} (n+m+1) - s \right)! \left( \frac{1}{2} (n-m-1) - s \right)!} r^{n-2s}, \text{ where } (\rho, \theta) \text{ is the polar coordinate of } (x, y). \]

Figure 3: (a), graphic illustration of the theoretical values of Zernike moments template (up to 4th order); (b–o), corresponding approximation using box filter.

The training images are then convolved with a 14-dimensional filter-bank created above. The 14D responses for all training pixels are then whitened (to give zero mean and unit covariance), and an unsupervised clustering is performed. We employ the Euclidean-distance K-means clustering algorithm, which can be made dramatically faster by using the techniques of [26]. Finally, each pixel in each image is assigned to the nearest cluster center, producing the texture extraction image. Note that for efficiency one can accelerate further using the \( kd \)-tree algorithm [27] to perform the nearest neighbor search. Figure 4a, 4b shows an example of the feature extraction process.
(2) Supervised training

Similar to [23], we adopted a supervised training algorithm based on both texture and spatial information. Each training sample is composed of a pair \((r, l)\) of an image region, \(r\), and a texture label (given by 14D clustering) \(l\). The feature response \(v\) at location \(i\) is the proportion of pixels under the offset region \(r+i\) that have texture label \(l\), and there will be zero contribution to the feature response outside of the region:

\[
v_{r,l}(i) = \frac{1}{\text{area}(r)} \sum_{j \in (r+i)} \left| \text{Texture label} = l \right|
\]

(6)

Which is illustrated in figure 4c. Looking at the rectangles in figure 4c reminds us of the old trick that has been played previously. Again, we employ integral map to accelerate the algorithm: If \(\hat{L}^{(l)}\) is the integral map for feature \(l\) (the filter response for \(l\)th Zernike moment), then the feature response can be computed as:

\[
v_{r,l}(i) = \hat{L}^{(l)}_{rb} - \hat{L}^{(l)}_{rl} - \hat{L}^{(l)}_{lb} + \hat{L}^{(l)}_{rl}
\]

(7)

where \(rb\), \(lb\), \(rl\) and \(rd\) denote the bottom right, bottom left, top right and top left corners of rectangle \(r\), respectively.

Inspired by [19], we intend to build a similar discriminative learning model. For this purpose, we employ an adapted version of the Joint Boost algorithm [28], which iteratively selects feature response as ‘weak learners’, and combines them into a strong classifier \(P(c|x, i)\), which forms a conditional random field model given the \(i\)th pixel in an image \(x\) as:

\[
\log P(c| x, \theta) = \sum_i \psi_i(c, x; \theta) = \log Z(\theta, x)
\]

(8)

Where \(c\) is the class label, \(P\) is the conditional probability with respect to \(c\), \(\theta\) is the model parameter, \(Z(\theta, x)\) is the partition function which normalizes the distribution, and \(\psi\) is the unary potential which is defined as:

\[
\psi_i(c, x; \theta) = \log(P(c, x, i))
\]

(9)

Joint Boost shares each weak learner between a set of classes \(C\), so that a single weak learner classify for several classes at once. This allows for classification with cost sublinear in the number of classes, and leads to improved generalization [28].

The learned strong classifier is an additive model of the form

\[
H(c, i) = \sum_{m=1}^{M} h^m(i)
\]

summing the classification confidence of \(M\) weak learners. The confidence value \(H(c, i)\) can be reinterpreted as a probability distribution over \(c\) using the soft-max or multiclass logistic transformation for:

\[
P(c|x, i) \propto \exp(H(c, i))
\]

(10)
Each weak learner is a decision stump based on feature response $v_{l,r}(i)$ of the form:

$$h_l(c) = \begin{cases} a[v_{l,r}(i) > \theta] + b & \text{if } c \in C \\ k^c & \text{otherwise} \end{cases}$$

(11)

With parameter $(a, b, \{k^c\}_{c \in C}, \theta, C, r, l)$. The region $r$ and label $l$ together specify the feature vector, and $v_{l,r}(i)$ denotes the corresponding feature response at position $i$.

For those classes that share this feature ($c \in C$), the weak learner gives $h_l(c) \in \{a + b, b\}$ depending on the comparison of $v_{l,r}(i)$ to a threshold $\theta$. For classes not sharing the feature ($c \notin C$), the constant $k^c$ ensures that unequal numbers of training examples of each class do not adversely affect the learning procedure. Each training example $i$ (a pixel in a training image) is paired with a target value $z_i^c \in \{-1,+1\}$ (+1 if example $i$ has ground truth class $c$, -1 otherwise) and assigned a weight $w_i^c$ specifying its classification accuracy for class $c$ after $m-1$ rounds of boosting. Round $m$ chooses a new weak learner by minimizing an error function $J_{wse}$ incorporating the weights:

$$J_{wse} = \sum_c \sum_i w_i^c (z_i^c - h_i^m(c))^2$$

(12)

The training examples are then re-weighted by:

$$w_i^c := w_i^c e^{-\varepsilon_i^c h_i^m(c)}$$

(13)

to reflect the new classification accuracy and maintain the invariant that $w_i^c = e^{-\varepsilon_i^c h_i(c)}$. This procedure emphasizes poorly classified examples in subsequent rounds, and ensures that over many rounds, the classification for each training example approaches the target value.

(3) **Online testing**

After the conditional random field model has been learned, the online testing step becomes as straightforward as looking for the most probable labeling $c^*$ that maximize the conditional probability of (8). An excellent detailed explanation of the algorithm can be seen in [29], and we briefly give the formula here for completeness:

Suppose that we have a set of labels $c$ and a fixed label $\alpha \in \{1, \ldots, C\}$, where $C$ is the number of classes. In the testing step, each pixel $i$ makes a binary decision: it can either keep its old label or switch to label $\alpha$. Therefore, a binary vector $s \in \{0,1\}^P$ can be introduced which defines the auxiliary configuration $c[s]$ as:
The algorithm computes optimal moves for labels $\alpha_i$ in orders, accepting the moves only if they increase the objective function. It has been proved that this algorithm is guaranteed to converge [29].

3 Results and Discussion

The proposed algorithm is implemented into an automated software system, and has been tested on clinical colonoscopic data. Sensitivity and specificity for this automated image analysis system were calculated for the normal colon tissues and detected abnormal findings using the colonoscopist’s annotations as the criterion standard. The true positive and true negative results were computed based on pixel to pixel match (number of overlapping pixels) between automated image analysis system algorithm results and colonoscopic annotations, the criterion standard (Figure 5). Therefore, sensitivity = area of true positives/(area of true positives + area of false negatives) and specificity = area of true negatives/(area of true negatives + area of false positives).

Two major parameters affect the performance of the algorithm (here we especially consider sensitivity as this is an application for fast screening): the highest order of Zernike moments we used in feature extraction and, the round of boosting in the learning step. Apparently, it is a trade-off between performance and efficiency due to limited computational resources. It can be seen from figure 6a that the performance gets better when more Zernike moments are selected in the feature extraction step. This is because higher order Zernike moments make more contribution to the description of detail texture. However due to truncated effect [24], we get minor improvement when the highest order is over four. Figure 6b shows the statistical analysis between sensitivity and round of boosting, when the round of boosting is more than 5000, the sensitivity curve begins to converge and we get best sensitivity around 80%.

$$c_i[s] = \begin{cases} c_i & \text{if } s_i = 0 \\ \alpha & \text{if } s_i = 1 \end{cases}$$

(14)
Figure 6. statistical analysis for sensitivity. (a) sensitivity with highest order of Zernike moments in the feature extraction step; (b) sensitivity with round of boosting in the learning step

We also compared the proposed algorithm with state-of-the-art [9, 23] (we appreciate the kindness of the authors to share their software) on a set of 103 clinical patient data, examined in regular check-ups with optical endoscope. All the data were then manually annotated by experienced colonoscopist, of which 30% of them are used as training dataset, while others used for later validation. The experimental results show that our proposed method visually performs well (see figure 7), and is both faster and performing better than Karkanis’s [9] and Shotton’s [23] method, especially under unknown rotation (when the training image and testing image are acquired with different viewpoint). This is easily understandable since our proposed method theoretically has two advantages: 1) the calculation of feature extraction and learning step is accelerated drastically using integral-map (in Shotton’s method they only utilized integral-map in the latter step); 2) The usage of Zernike moments guaranties rotation invariance.

Fig.7. performance of proposed algorithm on clinical colonoscopic data, (a) input images, (b) classification results (red areas represent normal tissue, green areas indicate the location of flat lesion)

Table 1 shows the quantitative comparison of the three methods on the average sensitivity, specificity and computational time. Future work will aim at optimizing the algorithm for additional speed up. Such implementation could be used during the colonoscopy to increase the physician’s capability to detect polyps faster, and thus reduce the long duration of the examination, which is both uncomfortable for the patients and inconvenient for the clinicians.
Table 1: quantitative comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specificity(%)</th>
<th>Sensitivity(%)</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karkanis’s method</td>
<td>14</td>
<td>15</td>
<td>90</td>
</tr>
<tr>
<td>Shotton’s method</td>
<td>50</td>
<td>73</td>
<td>4</td>
</tr>
<tr>
<td>Proposed method</td>
<td>65</td>
<td>80</td>
<td>3</td>
</tr>
</tbody>
</table>

4 Conclusion

Texture-based analysis and classification plays an important role in the screening and early detection of colorectal cancer, and researchers have investigated hard during the decade. However this application has been hindered by slow speed, due to lack of efficient algorithms. Inspired by Viola’s [19] and Bay’s [20] algorithms of using integral map, we further applied this technique into fast texture classification and segmentation. This paper presents an entire scheme of image classification, including feature extraction, supervised training and testing, which outperforms the current state-of-the-art, both in speed and accuracy. Zernike moments are also employed in the feature extraction to take advantage of their good properties of rotation invariance and orthogonality. The experimental results show visually over-all good performance in the real colorectal images, and great coherence with doctor’s annotation.

Acknowledgments. This paper is funded by Guangdong Natural Science Foundation (code:8178922035) and Shenzhen public research project (code:SY200806300212A).

References