Real-Time Phase Boundary Detection for Colonoscopy Videos Using Motion Vector Templates

Ruwan Nawarathna[1], JungHwan Oh[1], Jayantha Muthukudage[1], Wallapak Tavanapong[2], Johnny Wong[2], and Piet C. de Groen[3]

[1] Department of Computer Science and Engineering, University of North Texas, 1155 Union Circle #311366, Denton, Texas 76203-5017, USA
{rdn0025, Junghwan.Oh,mjk0129}@unt.edu
[2] Department of Computer Science, 226 Atanasoff Hall, Iowa State University, Ames, Iowa 50011-1040, USA
{tavanapo,wong}@cs.iastate.edu
[3] Department of Medicine, Division of Gastroenterology and Hepatology, Mayo Clinic, 200 First St. S.W., Rochester, Minnesota 55905, USA

Abstract. Colonoscopy is the preferred screening method currently available for detection of colorectal cancer and its precursor lesions, colorectal polyps. However, recent data suggest that there is a significant miss rate for the detection of polyps in the colon during colonoscopy. Therefore, techniques for real-time quality measurement and feedback are necessary to aid the endoscopist towards optimal inspection to improve the overall quality of colonoscopy during the procedure. A typical colonoscopy procedure consists of two phases: an insertion phase and a withdrawal phase. One of the most essential tasks in real-time fully automated quality measurement is to find the location of the boundary between insertion and withdrawal phases. In this paper, we present a method based on motion vector templates to detect the phase boundary in real-time. The proposed method detects the phase boundary with a better accuracy and a faster speed compared to our previous method.

Keywords: Colonoscopy, phase boundary, end of insertion, motion vectors, camera motion estimation, and motion vector templates.

1 Introduction

Colonoscopy is the preferred screening modality for prevention of colorectal cancer---the second leading cause of cancer-related deaths in the US [1]. A typical colonoscopy procedure consists of two phases: an insertion phase and a withdrawal phase. The main purpose of the insertion phase is to reach the end of the colon, whereas in the withdrawal phase, careful inspection of all visible mucosa, tissue sampling, polyp removal, etc., are performed. Despite being the preferred screening modality, recent data suggest that there is a significant miss-rate in the detection of even large polyps during colonoscopy [2]. The miss-rate may be related to the experience of the endoscopist and the location of the lesion in the colon, but no prospective studies related to this have been done thus far. The American Society for Gastrointestinal Endoscopy

H. Yoshida, D. Hawkes, M.W. Vannier (Eds.): Abdominal Imaging 2012, LNCS 7601, pp. 116–125, 2012.

has suggested many guidelines for best practices in colonoscopy as described in [2] which includes the duration of the withdrawal phase, the average polyp detection rate, and the thorough of inspection of the colon mucosa. In [3], six quality metrics are proposed which are based on the durations of the insertion and the withdrawal phases. Therefore, accurate detection of the phase boundary (end of insertion (*EOI*)) between the insertion phase and the withdrawal phase is very essential in fully automated quality analysis of colonoscopy procedures.

The best way to detect the *EOI* is to analyze the motion of the colonoscopy camera, specifically, the z-directional motion (i.e., dolling camera motion (*DCM*)). The reason is that the colonoscope moves in forward and backward directions inside the colon during colonoscopy [2]. An accurate estimation of the camera motion can be obtained by analyzing the change that occurred between two consecutive images (frames) (i.e., a frame pair) in the video due to the movement of the camera. This change can be represented by using motion vectors. A motion vector represents the displacement of an area (usually a macroblock) that occurred due to the movement of the camera. The major challenges in accurately detecting the EOI with this approach are (1) colonoscopy frames have various artifacts such as out of focus (i.e., blurriness), specular reflection, stool, and water which can make the motion vector generation process imprecise, (2) traditional camera motion models such as the affine model [4] produce inaccurate motions often for colonoscopy, and (3) the method must complete all tasks in real-time. Specifically, our colonoscopy videos output 30 frames per second in MPEG-2 format. So, in order to achieve real-time processing, motion estimation of a frame pair must be completed within 66 milliseconds (ms) (i.e., 33ms per frame x 2).

In this paper, we propose a new method to detect the *EOI* based on motion vector templates. This method attempts to analyze the motion vector distribution in the four corners of the frame pair in order to predict the camera motion. So, proposed method does not depend on heavy computations as in traditional models such as the affine model [4]. The motion vectors are generated using an optical flow block matching algorithm [5-7] which is more suitable for the motion vector generation in colonoscopy frames. Motion vectors are obtained only on the frames that guarantee to provide accurate motions. Therefore, the primary contributions of this paper are (1) we propose a new algorithm to estimate camera motions in colonoscopy videos more accurately using motion vector templates and (2) our new algorithm offer a very large improvement in the speed which leads to significantly better real-time *EOI* detection of colonoscopy videos compared to our previous method [4].

The remainder of this paper is organized as follows. Related work in the field of colonoscopy frame processing and a brief analysis of our previous work [4] are presented in Section 2. The proposed phase boundary detection technique is described in Section 3. In Section 4, we discuss our experimental setup and results. Finally, Section 5 presents some concluding remarks.

2 Related Work

The related works on colonoscopy can be divided into three main categories: (1) processing of frames for tasks such as non-informative frame detection, stool frame

detection, and many others [8-10], (2) detection of abnormalities such as colorectal polyps [10], and (3) analysis of the quality of colonoscopy procedures [3].

The only work on real-time *EOI* detection that can be found in the literature is our previous work outlined in [4]. In that work the *EOI* is detected by applying a three step approach; (1) motion vector generation using color-based block matching, (2) camera motion estimation using the affine model, and (3) accumulation of *DCM* values. Due to the following issues, the accuracy of the *EOI* detection in our previous work is not satisfactory. Color-based block matching for motion vector generation is heavily dependent on the color information of the images. But, the colonoscopy frames have a limited color range [1, 2]. Also, since colonoscopy frames have a variety of artifacts such as blurriness, stool, and water, color-based block-matching method generates many flawed motion vectors. Moreover, the affine camera model [4] is very sensitive to outliers and generates incorrect *DCM* values when diverse motion vectors are present. Due to these incorrect *DCM* values, unnecessary local maxima (i.e., peaks) will be generated during *DCM* accumulation; consequently an incorrect point is detected as the phase boundary. The proposed method overcomes these issues and detects the *EOI* with 22% better accuracy and with 40-times better speed when compared to the previous work as described in Section 3.

3 Proposed Method

The proposed method has four main steps: (1) preprocessing of colonoscopy frames to discard/enhance unsuitable frames for motion vector generation, (2) motion vector generation using optical flow, (3) camera motion estimation using motion vector templates, and (4) detection of the phase boundary by analyzing cumulative *DCM*s. In the following sections each step will be discussed in detail.

3.1 Frame Preprocessing

As mentioned in Section 1 various artifacts such as blurriness, specular reflection, stool, water, therapeutic instruments, etc in colonoscopy frames can create errors in the motion vector generation process. Hence, preprocessing of frames is mandatory to get accurate motion vectors. If at least one frame of a frame pair in a video stream is a blurry frame then that frame pair is ignored from the motion vector generation process using the method outlined in [8]. Stools can be found in most sections in the colon. By their nature, these stools can float inside the colon and generate object motions which can combine with the camera motions. This can provide very unreliable motion vectors. Therefore, we eliminate the frame pairs having at least one frame with a stool percentage greater than a certain threshold (set to 50% based on experiments) using the technique proposed in [9]. In addition to these frame pairs, we discard frame pairs if the frames in the frame pair are not sufficiently correlated (see Fig. 1(a)). We calculate a correlation score between the two frames in each frame pair as expressed in equation (1). Then, only the frame pairs having correlation scores within a threshold value range are selected for motion vector generation. We set the threshold range to 0.89 - 0.99. An upper threshold is used to remove highly correlated frames as such frame pairs produce very few motion vectors since they are very similar.

$$Corr(A,B) = \frac{E[(I_A - \mu_A)(I_B - \mu_B)]}{\sigma_A \sigma_B}. \quad (1)$$

In equation (1), I_A and I_B are the intensities; μ_A and μ_B are the mean intensities; and σ_A and σ_B are the standard deviations of intensities of consecutive frames A and B, respectively. E is the expected intensity value. Moreover, colonoscopy frames have significant amount of noise. Also, in some frames, the brightness is inconsistent across the frame and in some frames the contrast is low. The noise is reduced by using a Gaussian filter of size 5x5 [7], and the brightness distribution is made to be uniform by applying histogram equalization [7]. Fig. 1(a) and 1(b) show a discarded and a retained frame pair for motion vector generation.

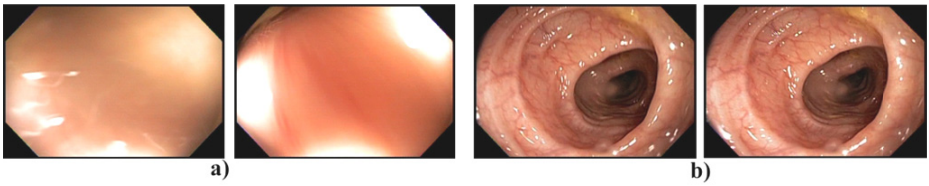

Fig. 1. a) A discarded frame pair because they are not sufficiently correlated, and b) a retained frame pair for motion vector generation

3.2 Motion Vector Generation Using Optical Flow Block Matching

We generate motions vectors on every 10^{th} frame pair in the video stream which pass the preprocessing stage. This is done because camera motion of 10 consecutive frames is very similar. Color-based block matching [4] and the optical flow [5-7] are the most widely used motion vector generation algorithms. Due to limited color information in colonoscopy frames [10] the accuracy of color-based block matching is not sufficient for motion vector generation. Numerous methods have been proposed to compute the optical flow from adjacent frames in a video. We tried a sparse optical flow method (iterative Lucas-Kanade method with pyramids [5]), a dense optical method (Horn and Schunck algorithm [6]) and a block matching based optical flow method [7]. Since colonoscopy frames lack distinctive features, the Lucas-Kanade method tends to select stool and specular reflection areas as feature points. Because stool and specular reflection areas produce object motions, most of the generated motion vectors by this method are related to object motions. The Horn-Schunck method produces a dense optical flow [6]. This method is computationally very expensive and it also requires a very good motion vector filtering algorithm since it can generate lots of dissimilar motion vectors. Due to these reasons iterative Lucas-Kanade and Horn-Schunk methods are not suitable solutions to our problem.

Optical flow block matching is known to be more precise than the global Horn-Schunck method, and may also be faster since no iterative scheme is needed. Also, this method ensures that the effect of object motions (created by stools and specular reflection) on motion vectors is minimized considerably. Thus, we decided to use the optical flow block matching algorithm to generate motion vectors. This method

attempts to divide both previous and current frames into blocks, and then computes the motions of these blocks using optical flow [5, 7]. For each $m \times m$ block (B_k) centered around pixel (x,y) in frame k, we obtain a search area S in frame *k-1* with B_k at the center block. The size of the search area S is $(m+2p) \times (m+2p)$ where p indicates the search range in pixels. Then, we compute the sum of square differences (*SSD*) between B_k and all possible $m \times m$ blocks in S as given in the equation (2). The $m \times m$ block (B_k') in S centered around (x',y'), which gives the lowest *SSD* is selected as the matching block. The displacement vector given by $u = x - x'$; $v = y - y'$ is the motion vector between B_k and B_k'. $f_k(x,y)$ is the intensity of the pixel at (x,y). We experimentally found that a block size of 8 × 8 pixels, a search area size of 16 ×16 pixels and a *SSD* threshold of 128 are able to generate more accurate motion vectors for colonoscopy. Fig. 2(a) shows a typical result from motion vector generation. (The reason for having motion vectors only in four corner regions is explained in Section 3.3).

$$(u,v) = \underset{\substack{u=0,\ldots,-p \\ v=0,\ldots,p}}{\operatorname{argmin}} \sum_{i=0}^{-m-1} \sum_{j=0}^{m-1} (f_k(x+i,y+j) - f_{k-1}(x+i+u,y+j+v))^2. \quad (2)$$

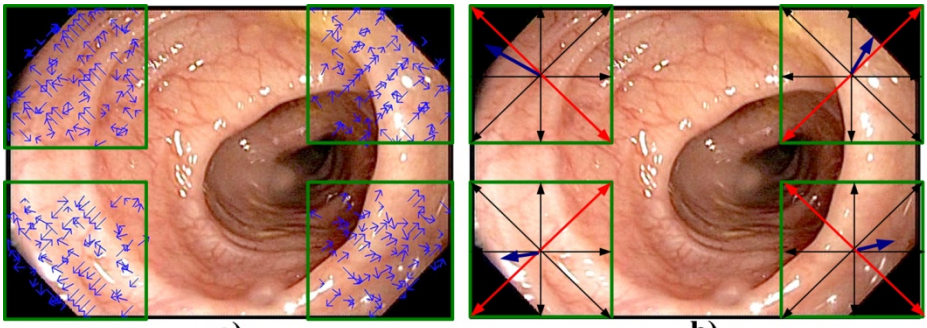

Fig. 2. a) Generated motion vectors from a frame pair in the four motion vector templates (arrow sizes are scaled up by a factor of 4) and b) their *TMV*s (i.e., blue arrows; *TMV*s are scaled so that they fit into their quadrants). Green squares represent the Template boundaries.

3.3 Camera Motion Estimation Using Motion Vector Templates

Motion vector templates have been studied previously in [11, 12]. A key aspect of our motion vector template method is the following: motion vectors in the four corners of our images show a unique pattern for each different type of camera motion. In the center part of the frame, we find either zero motion vectors (i.e., zero magnitude) (e.g. Z-directional translation) or motion vectors similar to four corners in the frame (e.g. X and Y motions). Hence, the camera motion can be estimated by analyzing the motion vector pattern in the four corners of a frame pair. Indeed, in the proposed method motion vectors are computed only in the four corners of a frame pair (see Fig. 2(a)). Four regions in the corners are called motion vector templates. Using our data set, we experimentally choose that the size of a Template as 13% of a frame, that is, 30% of

the width and 44% of the width of a frame (i.e., a Template is a square). Since, our aim is to find a specific pattern in the four motion vector templates; we represent the net motion in each Template region by one vector. This vector is computed by performing the vector addition of all motion vectors in each Template and then by calculating the mean vector as defined in equation (3). In a Template, the resultant vector is called its "Template Motion Vector (*TMV*)". In equation (3), mv_i^k is the k^{th} motion vector of the i^{th} Template and n_i is the number of motion vectors in the i^{th} Template. Four *TMV*s can be seen as illustrated in Fig. 2(b).

$$TMV_i = \frac{\sum_{k=1}^{n_i} mv_i^k}{n_i}. \tag{3}$$

Estimating the *DCM* by Computing the *DCM* Contribution. Forward and backward motions of the colonoscope can be estimated from positive *DCM* and negative *DCM*, respectively. Directions of the majority of motion vectors in a typical positive *DCM* are normally pointing from center to border. Hence, the directions of the four *TMV*s are also pointing from center to border (see Fig 3(a)). The directions of the majority of motion vectors and *TMV*s follow the opposite direction in a typical negative *DCM* (see Fig 3(b)). In general for a zooming motion, a similar behavior can be observed. Since, there is no zooming function available in a colonoscope; this pattern of *TMV*s can only be noticed when there is a *DCM*. Hence, we estimate the *DCM* when *TMV*s follow this pattern. We estimate the total *DCM* by calculating the *DCM* contribution from each Template. *DCM* contribution of a Template is calculated as the cosine of the angle that the *TMV* makes with the *DCM* axis multiplied by magnitude of the *TMV*. Use of cosine of the angle of *TMV*s ensures that more weight is assigned to the *TMV*s that are closer to the *DCM* axis (i.e., red line in each Template of Fig. 3). The rationale behind this design is that for a perfect *DCM* as seen in Fig. 3(a) and 3(b), *TWV*s are aligned with the *DCM* axis and the average of the magnitudes of *TMV*s represents the amount of *DCM*. For a non-perfect DCM as seen in Fig. 3(c), *TMV*s are away from the *DCM* axis and hence only a fraction of the magnitude represents the *DCM*. We compute the *DCM* contribution of a *TMV*, if it falls within the 45° range (in both directions, i.e., light blue and light green regions in Fig. 3) from the *DCM* axis. That means that a *TMV* within the 45° range from the *DCM* axis provides information about movement along the colon axis. *TMV*s which fall outside the 45° range carry essentially no information related to *DCM*. For this reason we ignore *TMV*s which fall outside the 45° region when calculating *DCM*.

The total *DCM* estimation process can be formulated as follows. We represent the magnitude and the angle of a k^{th} *TMV* which provides a non-zero positive *DCM* contribution as L_{pd}^k and θ_{pd}^k, respectively (see Fig. 3(a), *T1*,*T3* and *T4*). Also, L_{nd}^k and θ_{nd}^k represents the magnitude and the angle of the k^{th} *TMV* which provides a non-zero negative *DCM* contribution (see Fig. 3(c), *T2*). Here, *pd* stands for positive dolling and *nd* stands for negative dolling. In Fig. 3(c), *T1* has a positive *DCM* contribution and its value can be calculated as $L_{pd}^1 \cos \theta_{pd}^1$ and *T2* has a negative *DCM* contribution and its value can be calculated as $-L_{nd}^1 \cos \theta_{nd}^1$ and so on. We calculate the total *DCM* of a frame pair by taking *DCM* support count into consideration. The positive *DCM* support count (*PDSC*) is defined as the number of non-zero positive *DCM*

contributions and the negative *DCM* support count (*NDSC*) is defined as the number of non-zero negative *DCM* contributions. To be considered as a *DCM*, one of *PDSC* or *NDSC* must be at least two (experimentally decided). If we encounter equal (two) *PDSC* and *NDSC*, then we assign a zero *DCM*. Based on this model, we compute the average of positive *DCM* contributions for positive *DCM*, and the average of the negative *DCM* contributions for negative *DCM* as defined in equation (4).

$$DCM = \begin{cases} \dfrac{\sum_{k=1}^{PDSC} L_{pd}^{k} \cos \theta_{pd}^{k}}{PDSC}, & PDSC > NDSC \text{ and } PDSC \geq 2, \\ -\dfrac{\sum_{k=1}^{NDSC} L_{nd}^{k} \cos \theta_{nd}^{k}}{NDSC}, & NDSC > PDSC \text{ and } NDSC \geq 2, \\ 0, & \text{Otherwise.} \end{cases} \tag{4}$$

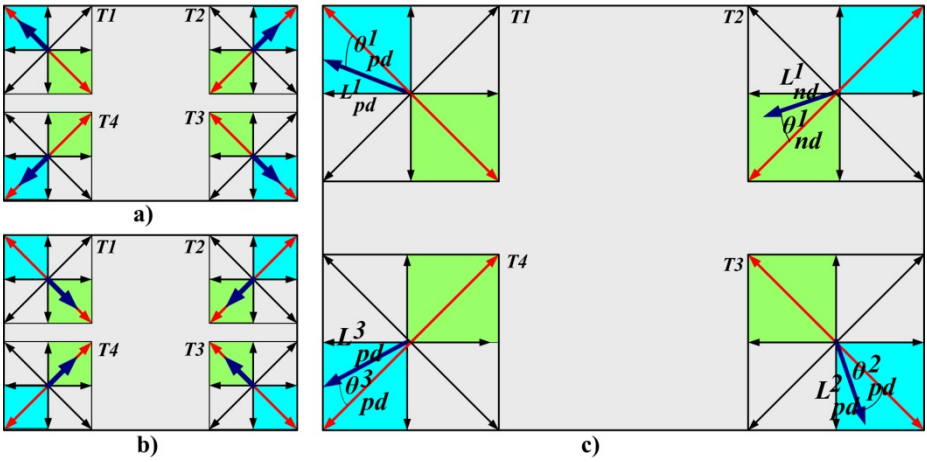

Fig. 3. An example of a) a perfect positive *DCM*, b) a perfect negative *DCM* and c) a *DCM* contribution calculation. Since *PDSC* > *NDSC*, (c) represents a positive *DCM* and it is also an example for a non-perfect positive *DCM*. In a), b), and c), *T1-T4* are four templates, red lines represent the *DCM* axes, light blue and light regions represent the positive *DCM* and negative *DCM* supporting regions, respectively. The dark blue arrows show *TMV*s.

3.4 Estimation of the Phase Boundary/End of Insertion (*EOI*)

The estimation of the *EOI* is done by analyzing the behavior of cumulative *DCM* (*CDCM*)) values of selected frame pairs in the entire procedure in real-time. We keep track of the *CDCM* values to find local maxima (i.e., peaks) in which the peak value remains unchanged for at least 1 minute (experimentally decided) as shown in Fig. 4. The frame number corresponding to this peak is assigned as the current *EOI* (i.e., a candidate *EOI*). Later, if we encounter another candidate *EOI* which has a *CDCM* value that is greater than the *CDCM* value of the current *EOI*, then the current *EOI* is updated with that candidate *EOI*. The frame number of the most recent candidate *EOI*

will be chosen as the *EOI*, which is the phase boundary. Obviously, the last *EOI* will be the frame number that defines the maximum *CDCM* value of the entire procedure. *CDCM* values of a full colonoscopy stream can be plotted as shown in Fig. 4. Since the size of a Template in the new method is 13% of a frame, we process only 52% of the frame area for motion vector generation. Also, we use an optimized version of the optical flow block matching algorithm provided by OpenCV 2.0 [7]. In addition to those, we apply CPU multithreading for motion vector generation by dividing the total number of blocks among the available processors for parallel processing. The combination of the above three factors ensures that the proposed method performs the *DCM* calculation well within real-time.

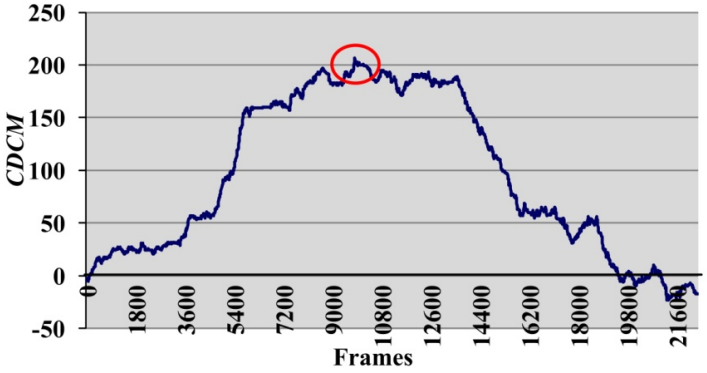

Fig. 4. A *CDCM* plot of a full colonoscopy video stream. The red oval is the detected EOI. Basically, this curve gives a snapshot of how the colonoscope moved through the colon in forward and backward directions during the procedure.

4 Experimental Setup and Results

The proposed method was implemented and integrated into SAPPAHIRE [13] which is a framework developed for real-time capture and quality analysis of colonoscopy. We perform our experiments in a simulation mode where videos are used as real-time video streams. All experiments were done in a computer with Intel(R) Core i7 2600K, 64-bit, 3.40GHz processor and 8 GB memory. We conducted experiments on ordinary videos as well as on real colonoscopy video streams.

We created a video which contains five types of motions. The video was recorded at a valley outside of a building. Five different types of motions are: (1) slow forward (frame 0 to 611), (2) fast forward (frame 741 to 1151), (3) fast backward (frame 1161 to 1431), (4) slow backward (frame 1441 to 2281), and (5) zero motion (frame 611 to 731 and frame 1091 to 1161). In the *CDCM* plot shown in Fig. 5, the fast motions (2 and 3) can be seen as steep lines, and slow motions (1 and 4) can be seen as less-steep lines. Also, zero motions (5) can be seen as straight lines. Therefore, our proposed method has accurately captured all motions that are contained in the video.

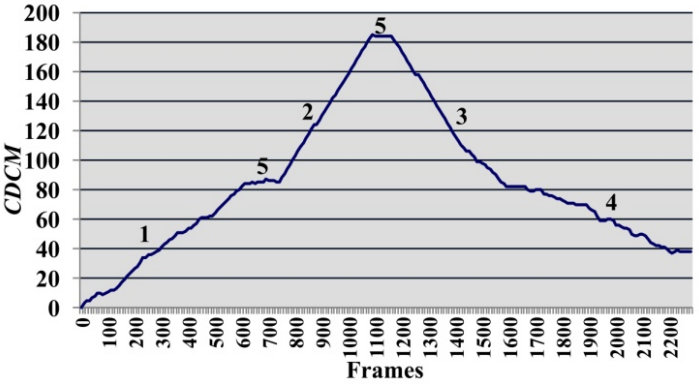

Fig. 5. CDCM plot of the ordinary video. Numbers 1-5 represent the five different types of motions present in the video.

Our real colonoscopy video set contains 146 videos. The average video length is 17.40 minutes and the frame size is 720 x 480. When the cecum is reached during colonoscopy, any remaining debris is removed followed by inspection of the appendiceal orifice, the ileocecal valve and if required or readily possible the distal terminal ileum. Clearing debris and inspection require about 1 to 2 minutes. Therefore, if the detected *EOI* falls within a 2 min range from the ground truth (i.e., before or after), then we consider it as a correctly detected *EOI* and vice versa. Outcomes from this experiment are summarized in Table 1. The results show that our new method detects the *EOI* with 82% accuracy. Therefore, compared to our previous method, we obtained a 22% improvement in accuracy. When comparing the average time difference with the ground truth of all videos of the two methods, the new method has shown a major improvement (81 seconds). Also, the proposed method provides a substantial gain (almost 40-times) in the average execution time per frame pair over the earlier method. The previous method satisfies the real-time constraint by a slight margin (by 13.70ms = 66 - 52.30). However, the new method satisfies the real-time constraint by a considerably larger margin (by 64.68 ms = 66 - 1.32).

Table 1. Effectiveness of the proposed phase boundary detection method

Description	Previous	New
Number of correctly detected videos	88	120
Accuracy of the *EOI* detection	**60%**	**82%**
Average time difference with the ground truth *EOI* (mm:ss)	02.54	01:33
Average execution time per frame pair (ms)	52.30	1.32

5 Conclusion

Finding the phase boundary is considered to be a very important task for fully automated quality analysis of colonoscopy. We have proposed and implemented a new algorithm for real-time phase boundary detection in colonoscopy using motion

vector templates. The proposed method detects the phase boundary with 82% accuracy which is 22% better than our previous algorithm. Also, the new algorithm is 40-times faster than our previous algorithm. Moreover, this method is easy to implement since it does not possess heavy computations compared to the previous method. Experimental results also demonstrate that this method works very well on estimating the dolling motions of ordinary videos. Accurate detection of the phase boundary leads to generation of many quality metrics, such as the ones proposed in [3]. A key problem affecting the accuracy of our current method is formed by frames containing remaining debris, air bubbles and water or specular reflections. In the future we will improve the accuracy by detecting and eliminating these frames.

Acknowledgments. This work is partially supported by NSF STTR-Grant No. 0740596, 0956847, National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK DK083745).

References

1. American Cancer Society. Colorectal Cancer Facts & Figures (2008), http://www.cancer.org/docroot/STT/content/STT_1x_Cancer_Facts_and_Figures_2008.asp
2. Rex, D.K., Petrini, J.L., Baron, T.H., Chak, A., Cohen, J., Deal, S.E., Hoffman, B., Jacobson, B.C., Mergener, K., Pertersen, B., Safdi, M.A., Faigel, D.O., Pike, I.M.: Quality Indicators for Colonoscopy. Gastrointestinal Endoscopy 63, S16–S26 (2006)
3. Oh, J., Hwang, S., Cao, Y., Tavanapong, W., Liu, D., Wong, J., de Groen, P.C.: Measuring Objective Quality of Colonoscopy. IEEE T. Bio-Med. Eng. 56(9), 2190–2196 (2009)
4. Oh, J., Rajbal, M.A., Muthukudage, J.K., Tavanapong, W., Wong, J., de Groen, P.C.: Real-Time Phase Boundary Detection in Colonoscopy Videos. In: 6th International Symposium on Image and Signal Processing and Analysis, Salzburg, pp. 724–729 (2009)
5. Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker. Intel Corporation Microprocessor Research Labs 1(2), 1–9 (2001)
6. Horn, B., Schunck, B.: Determining Optical Flow. Artif. Intell. 17, 185–203 (1981)
7. Bradski, G., Kaehler, A.: Learning OpenCV:Computer Vision with the OpenCV Library. O'Reilly Media (2008)
8. Oh, J., Hwang, S., Lee, J., Tavanapong, W., Wong, J., de Groen, P.C.: Informative Frame Classification for Endoscopy Video. Med. Image Anal. 11(2), 110–127 (2007)
9. Muthukudage, J., Oh, J., Tavanapong, W., Wong, J., de Groen, P.C.: Color Based Stool Region Detection in Colonoscopy Videos for Quality Measurements. In: Ho, Y.-S. (ed.) PSIVT 2011, Part I. LNCS, vol. 7087, pp. 61–72. Springer, Heidelberg (2011)
10. Atasoy, S., Mateus, D., Lallemand, J., Meining, A., Yang, G.-Z., Navab, N.: Endoscopic Video Manifolds. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part II. LNCS, vol. 6362, pp. 437–445. Springer, Heidelberg (2010)
11. Nguyen, T., Laurendeau, D., Branzan, A.: A Robust Method for Camera Motion Estimation in Movies Based on Optical Flow. Int. J. Intell. Syst. Technol. Appl. 9(3/4), 228–238 (2010)
12. Xiong, W., Lee, J.C.: Efficient Scene Change Detection and Camera Motion Annotation for Video Classification. Computer Vision and Image Understanding 71(2), 166–181 (1998)
13. Stanek, S.R., Tavanapong, W., Wong, J., Oh, J., Nawarathna, R., Muthukudage, J., de Groen, P.C.: SAPPHIRE Middleware and Software Development Kit for Medical Video Analysis. In: 24th IEEE International Symposium on Computer-Based Medical Systems, Bristol, pp. 1–6 (2011)