Global motion compensation for respiration monitoring using hand-held camera

Zhang, Q.

Award date:
2012

Master Thesis

Global Motion Compensation for Respiration Monitoring using Hand-held Camera

Author:
Qiaowei Zhang

Committee:
Prof. dr.ir. Gerard de Haan (TU/e, Philips Research)
Dr.ir. Ihor Kirenko (Philips Research)
Dr.ir. Sveta Zinger (TU/e)

August 9, 2012

Acknowledgements

First of all, I would like to express my gratitude to my supervisors, Prof.dr.ir.Gerard de Haan, Dr.ir.Ihor Kirenko and Dr.ir.Caifeng Shan, for their consistent and sincere helps during the entire thesis project. I also would like to thank all the colleges in the Video & Image Processing group, Philips for their assistances and opinions about my presentation. After working with them for nearly seven months, I have learned a lot not only about the techniques but also more importantly, how to be professional engineer and researcher as they are.

A gratitude should be conveyed to all the teachers and classmates who have helped me during the master project in Technology University of Eindhoven. It is a pleasure to spent two years with them.

Thanks for my family in China. They are the internal motivation through my entire life, and they always support me and make me confident when I encounter troubles. Although they can not stay with me in the daily life, but their words are most powerful to me. A very special thanks to my girlfriend, Xizhu. Her love and care can not be expressed into words.

For all the pleasure during two years in Netherlands, it is thankful that I made the right choice to come here at that time. And I hope everything is getting better and better in the future.

Abstract

As one of the vital diagnostic signals, respiration rate is monitored for the purpose of patient surveillance in the hospital, since it can predict potentially serious clinic events. Conventionally, many contact-based methods are used, such as EBD-Derived Respiration or pneumograph. Compared with contact-based respiration monitoring, non-contact methods have advantages in: 1) Unobtrusiveness. Subjects are absent from distress caused by a contact device to avoid interferences of the measurement. 2) Comfort. Sensors attached may irritate the subject's skin, especially for long time surveillance. 3) Convenience. It is convenient for subjects without consideration for sensors and even connected wires. Furthermore, this technique can be implemented with a camera in the hand-held device, for instance, a smart phone. Unfortunately, currently reliable optical measurement of respiration is sensitive to all kinds of camera movements. This thesis provides a motion compensation algorithm for the video-based respiration measurement. Compared with existing method, our algorithm improves PSNR by the gain of 1.7747, and improves correlation metrics by 0.8063. Inevitably, there is always residue motion in the stabilized video, and those residue motion can decrease the accuracy of respiration measurement algorithm. Therefore, a post-processing algorithm is proposed to further improve the motion robustness. By adding missing peaks and deleting wrong peaks in the respiration curve, the post processing algorithm can further improve the accuracy of respiration measurement according to our benchmark.

Keywords: Respiration monitoring, hand-held camera, global motion compensation

Contents

Chapter 1

Introduction

Respiration in physiology refers to the transport of oxygen from the outside air to the cells within body tissues, and the transport of carbon dioxide in the opposite direction. Respiration rate (RR) is the number of breaths taken within a certain amount of time, typically one minute. As one of the vital diagnostic signals, respiration rate is mainly monitored for the purpose of patient surveillance, since it can predict potentially serious clinic events, such as apenea. The most intuitive way to measure respiration rate is to count by a person how many times the chest rises in one minute, which is inaccurate and unrealistic for long-time surveillance. Along with breathing behaviour, many phenomenons can be observed, for instance, respiratory sounds, respiratory airflow and respiratory chest movement. Based on measuring one of the above parameters, many respiration monitoring methods have been proposed.

1.1 Contact-based Respiration Monitoring

Currently, standard methods for respiration monitoring are all contact-based and non-invasive, including ECG-Derived Respiration (EDR) and pneumograph. The ECG-Derived Respiration (EDR) technique is based on the facts that 1) the positions of ECG electrodes on the chest surface move relative to the heart, and 2) transthoracic impedance varies, as the lungs fill and empty [7]. Commercial ECG monitor is usually integrated with respiration monitoring. The respiration can also be monitored by pneumograph that measures the chest and abdomen movement. There are various kinds of pneumograph, depending on different principles of operation. Conventionally, a thick rubber of elliptical shape is attached to the chest. Nowadays, electrical impedance is inserted into the elastic chest belt. During the respiration process, the expansion of the chest will change the impedance of the sensor. By measuring the current flows through the belt, the respiration signal can be extracted.

(a) electrodes in ECG

(b) chest strap

Figure 1.1: Currently standard respiration monitoring: a) ECG, b) pneumograph

Considering the electrodes and the chest belt (Figure 1.1), contact-based respiration monitoring requires a direct contact with the subject's body, which can irritate the skin, and cause skin to grow itchy or become sore, especially if the device is worn for a long period of time. Moreover, it is obtrusive in daily life, considering extra efforts required to wear on and off the device, and may also influence the accuracy of the measurement because of the subject's awareness of the sensor.

1.2 Non-contact Respiration Monitoring

As a unobtrusive way to monitor the subject, non-contact respiration monitoring attracts more and more attentions recently. Compared with contact-based respiration monitoring, non-contact methods have advantages in: 1) Unobtrusiveness. Subjects are absent from distress caused by a contact sensor to avoid interferences of the measurement. 2) Comfort. Sensors attached may irritate the subject's skin, especially for long time surveillance. 3) Convenience. It is convenient for subjects without consideration for sensors and even connected wires.

Non-contact methods are based on measurement of chest and abdomen movement caused by respiration. Instead of using expensive and cumbersome radar-based equipment [12], a basic digital camera such as a webcam can further provide portability and low cost. Philips Electronics B.V. developed a respiration measurement algorithm, referred as ProCor [13] based on optical detection of chest movement. However, ProCor is sensitive to all kinds of motions, including local motion of the subject, and the global motion induced by the camera. The limitation of this method is that the breathing signal is likely to be corrupted by motion. Non-contact respiration monitoring can perform well if no motion corruption occurs. But it is certainly not the case with hand-held camera. Generally speaking, the chest and abdomen movement caused by respiration is relatively small local motion, compared with large global motion induced by camera movement. It is a challenge to extract the small local motion from relatively large global motion using image and video processing techniques.

1.3 Respiration Monitoring in Using Hand-held Camera

In this thesis, we consider the problem related with respiration monitoring using a hand-held camera. Generally speaking, the camera, as a rigid object with certain size, can have two types of motions corresponding to two coordinate systems (Figure 1.2). All these motions

in 3-D space can be projected onto the frame plane, and affect the video recorded by the camera. On one hand, the camera can move freely in 3-D space. Movement in Z direction causes scaling in the video, while the movement in X or Y direction contributes to translation. On the other hand, the attitude of the camera can alter in the coordinate system with the camera itself as the origin. The attitude alternations include yaw, pitch and roll. Roll of the camera causes rotation in the frame, and the effect of yaw and pitch can be considered as translation with perspective distortion. Since yaw and pitch change the viewing angle from the camera, the frame plane is no longer parallel to the camera.

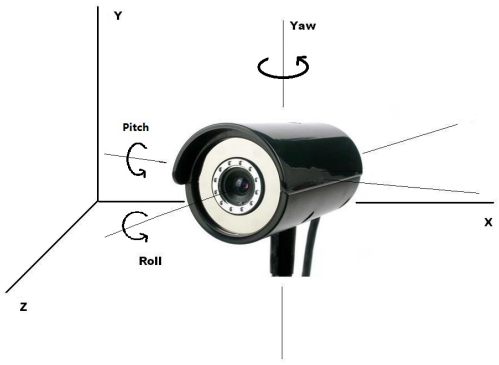


Figure 1.2: The possible types of camera movements in 3-D space.

Respiration signal extracted from local motion is relatively small compared with global motion induced by the camera movement. There are three basic types of effects of the global motion on the frame: (1) **Translation**: Camera movement in XY plane or yaw and pitch can cause translation in the frame, denoted by the amount of shift (t_x, t_y). (2) **Rotation**: Roll of the camera can cause rotation in the frame, denoted by rotation angle θ. (3)**Scaling**: Camera movement in Z plane will contribute to scaling effect in the frame series, denoted by scaling ratio R_s. Furthermore, yaw and pitch will also contribute to perspective distortion because the distance between the camera and the object is different for different viewing angle. But we assume that people who holds the camera intend to focus on the subject, so both the range and variation speed of translation (t_x, t_y), rotation angle θ and scaling ratio R_s are limited, and the perspective distortion caused by yaw and pitch can be ignored.

The report is structured as follows. In Chapter 2, original respiration algorithm ProCor is presented as well as the existing compensation algorithm. In Chapter 3, our compensation algorithm is proposed to improve motion robustness. In Chapter 4, a post stabilization algorithm is invented to further improve the accuracy of respiration measurement. Finally in Chapter 5, Summary is given.

Chapter 2

Respiration Measurement Algorithm

A camera-based algorithm (ProCor) [13] has been developed at Philips Research Eindhoven to meet the demand for unobtrusive monitoring of respiration signal. Unlike other camera-based algorithm [10], no markers are required to enhance visibility of the subtle respiration motion. The algorithm detects the small local motions of chest of a subject induced by respiration, and calculates the respiration rate from the frequency of those small motions. A region of interest(RespROI) (Figure 2.1) is required as an input for the respiration measurement algorithm to indicate where the chest of the subject is. The method is further intended to be applied on mobile devices to improve portability. Unfortunately, the existing algorithm is not robust to all kinds of motions. During the measurement, the camera has to be static and the subject has to be stationary. This is unrealistic when the camera is located in a hand-held device. In this chapter, we will introduce the details of ProCor and the existing compensation algorithm.

2.1 ProCor

An overview of the respiration measurement algorithm is illustrated in Figure 2.2. The algorithm consists of a Profile Correlation(ProCor) algorithm to derive respiration curve from the chest movement and a post processing algorithm to calculate the respiration. The system first reads a frame from either a camera or from a recorded video in the memory. A ROI is manually selected by users to indicate the chest of the subject at the beginning. Then ROI of each frame is projected to vertical axis to create a 1D profile, since horizontal motions of the chest are more likely polluted by other unintentional movement. Shift between two consecutive profiles, calculated by correlation, indicates changes of the chest position. So the raw respiration signal can be reconstructed by integrating the shift over time. (Figure 2.3).

Figure 2.1: RespROI and NoMeasureROI displayed in the frame: the white rectangle is RespROI, as the original input of respiration measurement algorithm(ProCor). The black rectangle is NoMeasureROI, input of the compensation algorithm proposed by van den Helder [27]

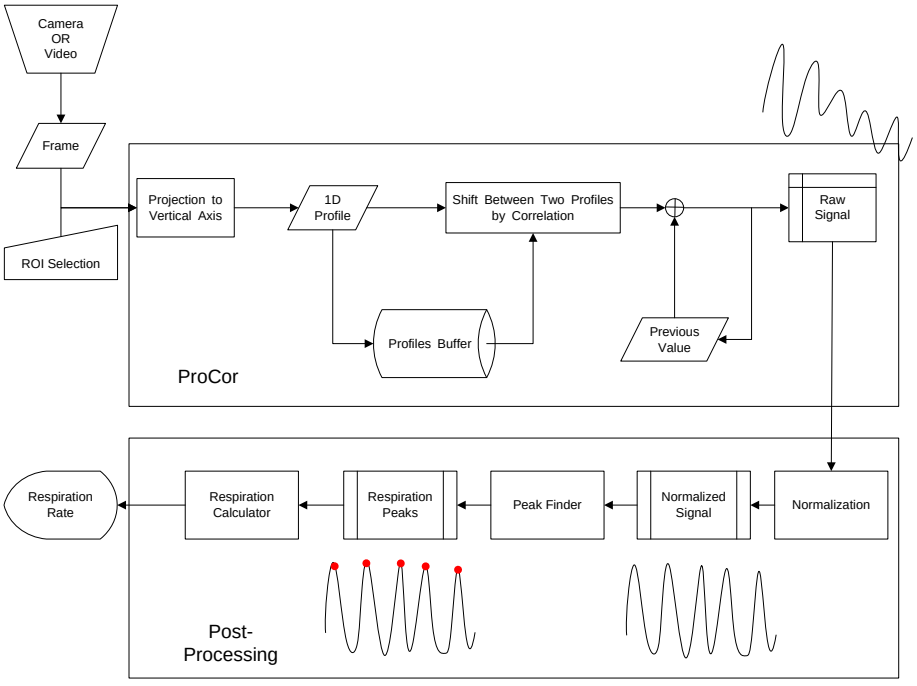


Figure 2.2: Overview of the respiration measurement algorithm

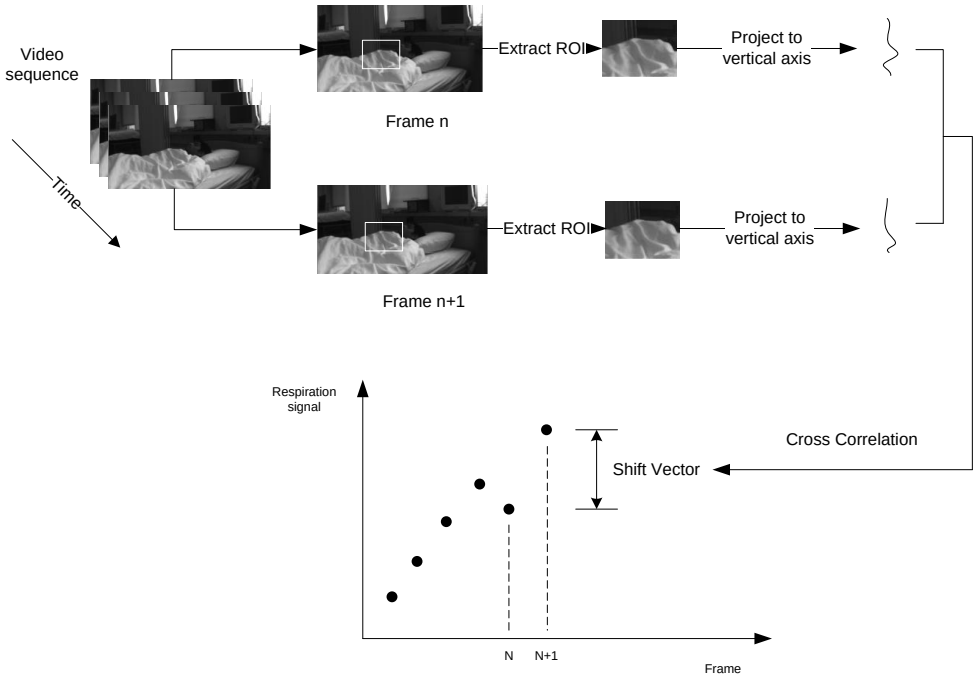


Figure 2.3: Extraction of ROI from a video sequence, and how respiration curve is integrated over time

Post-processing calculates the respiration rate from the raw respiration signal. The raw respiration signal is restored in a circular buffer, and first normalized according to current maximum and minimum value.

$$S_{norm} = S_{raw}/(Max(S_{raw}) - Min(S_{raw})) \tag{2.1}$$

Peak detection algorithm finds peaks on normalized signal. The respiration rate is mainly influenced by the peak detection algorithm [13]. Hereby is the pseudo-code of peak detection algorithm applied in the project.

Algorithm $PeakDetect(S = p_1, p_2, p_3..., p_n)$
1. n ←number of the points in S
2. dev ←deviation of the points in S
3. min ←$Float_{max}$; max ←$-Float_{max}$
4. minPosition ←-1 ; maxPosition ←-1
5. FlagForMax ←1
6. **for** i ←1 **to** n
7. **if** $p_i < min$
8. $min = p_i$
9. $minPosition = i$
10. **else if** $p_i > max$
11. $max = p_i$
12. $maxPosition = i$
13. **if** FlagForMax

14. **if** $p_i < max - dev$
15. p_i is a peak
16. $min = p_i$
17. $minPosition = i$
18. $FlagForMax = 0$
19. **else** **if** $p_i > min + dev$
20. $max = p_i$
21. $maxPosition = i$
22. $FlagForMax = 1$
23. **return** all the peaks and their positions

In order to avoid local maximum, the peak detection algorithm uses deviation of all available signals to determine whether it is a peak. So it is vital to implement normalization before peakdetection algorithm to avoid the impact from DC signal in the respiration curve.
With peaks in respiration signals, respiration rate is calculated by a window function to count at least three peaks. A window-based calculation can respond to changes more quickly. Assume the number of respiration signal points in the window is m, and the number of peaks found in the window is p, then respiration rate per minute (RR/min):

$$RR/min = 60 * p * framerate/m \tag{2.2}$$

2.2 Existing Compensation Algorithm

Van den Helder [27] proposed a compensation algorithm to stabilize the video while using an extra NoMeasureROI (Figure 2.1) to protect the respiration signal. RespROI is the input of respiration measurement algorithm ProCor, to indicate the position of subject's chest. And NoMeasureROI is the input of compensation algorithm, such that all pixels outside the NoMeasureROI will not have any respiration signals. So the compensation based on the part outside the NoMeasureROI will not ruin the subtle respiration signal. The idea behind the existing compensation algorithm is to first stabilize the video from hand-held camera as a pre-processing, then use ProCor and post-processing algorithm to calculate the respiration rate from stabilized video. Ideally, if the video is fully compensated, original ProCor algorithm will work. However, the stabilization algorithm is limited to only compensate translation.
Several estimation algorithms are implemented to estimate motion vectors, from which a global motion vector is derived by either mean or median method. The limitation of the method lies in using a unified motion vector to describe the global motion. Therefore, the global motion model is only a translation model. That is why only translation can be compensated.

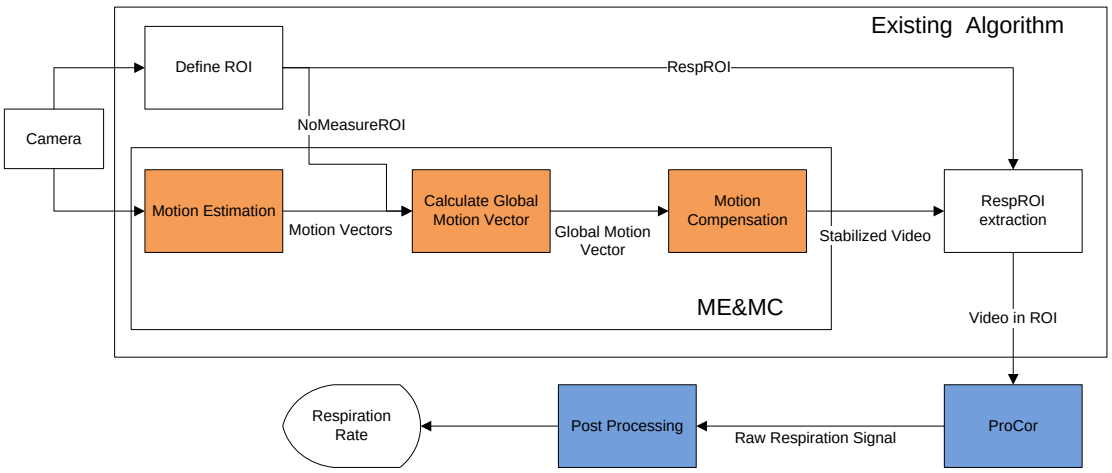


Figure 2.4: Overview of existing compensation algorithm.

In the next chapter, our stabilization algorithm is proposed to deal with all affine transformation caused by the global motion.

Chapter 3

Motion Estimation and Compensation

In this chapter, we try to improve the motion robustness by a global motion estimation and compensation algorithm. First, general concept of global motion estimation and compensation is introduced. Various algorithms have their own advantages and disadvantages. We have chosen a feature-based algorithm Lucas-Kanade with affine model, since the combination is most suitable for our scenario.

3.1 General Concept

Global motion estimation(GME) and compensation(GMC) (Figure 3.1) are two essential parts to stabilize the video sequence: 1) GME provides motion parameters for global motion models by analysing the input video; 2) GMC uses the global motion model to reconstruct the stabilized video from input video. In this section, we briefly review previous works about global motion estimation, and compare different global motion models for compensation.

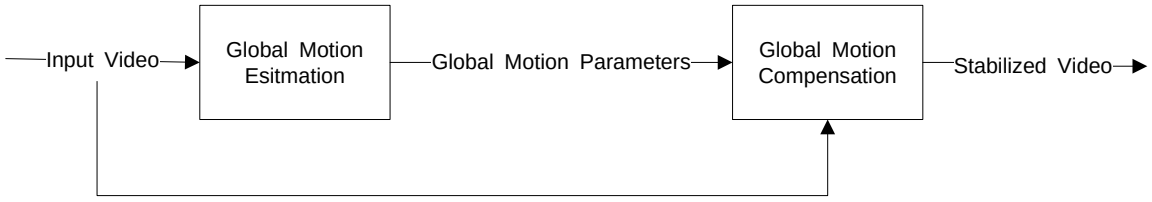


Figure 3.1: Block Diagram of GME and GMC

3.1.1 Global Motion Estimation

Global motion estimation(GME) algorithms can be classified by the size of units. Most motion estimation algorithms directly deal with pixels, blocks(group of pixels), objects(groups of pixels sharing a similar motion) or features(salient pixels with large gradients).

Pel-recursive algorithms(or gradient algorithms, pixel based) improve earlier ME algorithm [3] by estimating difference between actual and the predicted displacement vector. The difference vector, called *update*, is calculated for each pixel. This method, first proposed by Netravali [20], iteratively minimizes the square of displaced frame difference(DFD) by the steepest descent technique. The conventional pel-recursive algorithm suffers from several drawbacks. First, the iterative process may converge to a local minimum rather than a global one. Second, the converge process is quite slow since it iterates from pixel to pixel towards the true displacement. Third, the computational complexity increase dramatically with the size of the frame.
Many improved algorithms have been addressed. Walker [28] extends the algorithm to improve the convergence rate substantially and a different scheme for using the motion estimate is presented which eliminates explicit address transmission. Biemond [1] assumes that the *update* and the linearisation error are samples of wiener processes and improves the algorithm in convergence speed and reduction of prediction error. Compared with the methods by Netravali [20] and Walker [28], the improved approach shows advantages in robustness, stability and convergence. Csillag[4] further improves the accuracy of Biemond's approach[1] by applying hierarchical motion estimation and reducing the image areas of slow convergence. Efstratiadis[8] presents an improved algorithm of Biemond's approach[1] by utilizing the spatio-temporal correlations of the image sequence by considering an autoregressive(AR) model for the motion compensated frames. Estrela[9] uses spatially-adaptive regularization to improve the Wiener-based pel-recursive algorithm [1] and employs generalized cross-validation(GCV) to determine the optimal value of the regularization parameter for each pixel.
In **block-matching algorithms**, every frame is divided into non-overlapping square blocks with size $N * M$. The motion vector is assigned to every block instead of each pixel by searching a similar block within certain search area in the previous frame. The similarity between blocks is determined by the match criterion, for instance, Summed Absolute Difference(SAD), Summed Squared Difference(SSD), Mean Squared Error(MSE) and etc. Besides the match criterion, the performance of block-matching algorithms also depends on the search strategy. Typical full search algorithm(FSA) searches the best matched block in the search area $(2W + N) * (2W + m)$ in the previous frame, where W is the size of the search area, usually is the maximum displacement. Despite the heavy computations it requires, FSA can guarantee global minimum. In order to decrease the computational complexity, many improved search strategies have been proposed. For instance, three step search(TSS)[14], new three step search(NTSS)[16], four step search(FSS)[21], block-based gradient descent search(BBGDS)[17], diamond search(DS)[30] and HEXagon based search(HEXBS)[29]. These fast search algorithms may suffer from the disadvantage that they may get stuck in local minimum using SSD or SAD as the matching criterion, causing aperture problem. The 3-D recursive search(3DRS) algorithm[6] is proposed to provide true motion estimation to solve the problem. Currently, 3DRS is one of the most popular algorithms widely applied in motion estimation considering true motion estimation and computational complexity.
In the standard block-matching algorithm, the motion is restricted to translation. However, the algorithm can be extended by adding an affine model to estimate rotation. In [11] an affine model is proposed, where each block performs an affine transform instead of a translation. Similarly, a generalized block matching algorithm [23] is proposed which can handle complex motions including rotation. Of course, more complex motion model leads to heavier computational workloads.

Object-based algorithms replace block-based motion vector with object-based motion vector. In block-matching algorithm, the same motion vector is assigned to all pixels in the square block, however, the edge of actual moving object can not always be square and coincide with the block. Then the performance of block-matching algorithm is quite bad at the boundary of the object. It is more reasonable to assign a motion vector to every individual object, since pixels in a segmented object indeed share the same motion vector.
Object-based algorithms([19], [25], [15]) are mainly limited in the research field, because they are computationally expensive and inaccurate compared with other estimation algorithms with regarding to object segmentation algorithm required in the first step.
Feature-based algorithms follow the same principle to estimate the motion by matching among consecutive frames. Instead of segmenting the frame spatially, the feature-based algorithms choose *features* to represent the whole frame. Therefore, it is extremely important to choose *good* features at the beginning. Those *good* features should be easily identified and tracked from frame to frame. Shi and Tomasi[24] provided a feature selection criterion based on dissimilarity that uses affine motion as the underlying image change model. The best tracking method is the one first proposed by Lucas and Kanade[18] in 1981, and many improvements have been made based on that. A pyramidal implementation of classical Lucas-Kanade algorithm[2] improved the local tracking accuracy and robustness. Tomasi and Kanade[26] proposed a selection criterion of feature windows to improve the tracking algorithm.
Compared with block-matching algorithm, feature-based algorithms are less sensitive to rotation, scaling and other perspective transforms, since feature points do not assume a uniform motion vector for each block. The algorithm is also available in OpenCV and open source(KLT tracker).

3.1.2 Global Motion Model

A parametric motion model is a mapping M(Figure 3.2) from the coordinates of the pixel in current frame $X' = (x, y, 1)^T$ to previous frame $X = (x', y', 1)^T$: $X' = MX$. Various models include two-parameter translation model, four-parameter rotation-scale-translation model, six-parameter affine model and eight-parameter perspective model.

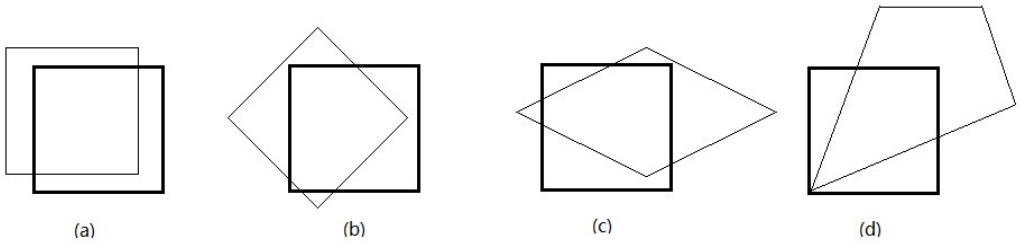


Figure 3.2: Example of various mapping functions: translation transform(a), RST transform(b), affine transform(c), perspective transform(d)

Translational Model

The mapping between current frame and previous frame is

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $[t_x, t_y]$is the translation offset. This model, used in previous work [27], can only describe translation.

RST Model

RST model is suitable to describe motions including rotation, scaling and translation. But the shape of the frame is preserved. The mapping matrix can be expressed as

$$M = \begin{bmatrix} r\cos\theta & -r\sin\theta & t_x \\ r\sin\theta & r\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

where r is the scaling ratio, θ is the rotation angel in degree, and vector $[t_x, t_y]$ is the translation in pixels.

Affine Model

A more complex model with six parameters is the most popular one, affine model. The affine model can describe any affine transformation. Parameter matrix A is related with scaling and rotation, and parameter vector b is the translation. Straight lines and the parallelism between straight lines are preserved in the affine transform, and it is assumed that the distance of the camera to the frame plane is not changed or the change can be ignored compared with the size of the frame. The mapping matrix

$$M = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

If $a_{11} = a_{22}$ and $a_{12} = -a_{21}$, the affine model can be reduced to four-parameter Rotation-Scale-Translation model (RST).

Perspective Model

Since the frame plane is not always parallel to the camera during the perspective transform, the assumption in affine model about the distance of the camera is not valid any more. The perspective mapping is

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

It is noted that only two parameters are independent among a_{31}, a_{32}, a_{33}, since they satisfy $a_{31}x + a_{32}y + a_{33} = 1$. Therefore, the perspective mapping requires eight parameters. Straight lines are preserved in the perspective transform. For example, a square can be mapped into the shape of lines connecting four arbitrary points.

3.1.3 Selecting Suitable Motion Estimation Algorithm and Motion Model

The motion estimation algorithm and motion model should be chosen based on user scenario. Our scenario includes: 1) people tend to focus the camera on the subject, but the unintentional movement is unavoidable. 2) subject can not move during the measurement. 3) the distance between the camera and the subject is 1 to 1.5 meters. Thus, it is a trade off between flexibility and complexity when choosing the motion model. The model with more parameters can describe more complex global motion, but also means more computational workloads and more valid feature points required to determine the parameters. The camera movement is not large according to scenario 1, and the difference of the viewing angle is limited considering small distance between the camera and the subject. Thus, perspective transform due to the changing viewing angle is not dominant in our scenario , therefore an affine model is selected.

As to the motion estimation algorithm, pixel-based algorithm suffers from large computational complexity, and metric-optimal method can not generate true motion vectors. Object-based algorithm is still in research area. Although it may have advantages considering segmentation of the subject from the background, the accuracy of currently available implementation is not good. Only block-based and feature-based algorithms are feasible. Block-based algorithm, especially true motion estimator, such as 3DRS, is computationally efficient in estimating true motion vectors for a block in a frame. However, it is designed for local motion estimation. For estimation of global motion vector, the size of the block can not be small, since it may be trapped by local motion. The size and number of the motion vector are also limited compared with feature-based algorithm. Therefore, feature-based algorithm is more suitable for global motion estimation. Among them, optical flow using Lucas-Kanade feature tracker is most popular, and computationally efficient implementation is available in OpenCV.

3.2 Optical Flow

Optical flow means tracking features in a frame across the video sequences. Actually human does optical flow all the time — unconsciously track moving objects by eyes. Formally, given point $[u_x, u_y]$ in frame F_1, and find the same point $[u_x + \delta_x, u_y + \delta_y]$ in frame F_2 that minimize the error ϵ:

$$\epsilon(\delta_x, \delta_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (F_1(x, y) - F_2(x + \delta_x, y + \delta_y)) \tag{3.1}$$

(δ_x, δ_y) is the optical flow, or motion vector between frame F_1 and frame F_2, and w_x, w_y are the width and height of the integration window. Typically, $\delta_x < w_x$ and $\delta_y < w_y$. Known as the aperture problem, the optical flow can not be solved unless additional conditions are added. Among all methods for optical flow estimation, Lucas-Kanade differential method [26] is the most popular. In addition, it is vital to select good features to track. Intuitively, a good feature should at least have two properties: 1)texturedness. Lack of texture means ambiguity in tracking. 2)corner. No corner contributes to the aperture problem. Shi [24] proposed a criterion to select good features.

Both good features to track and Lucas-Kanade feature tracker have been implemented In OpenCV. *cvGoodFeaturesToTrack*() and *cvCalcOpticalFlowLK*() are always combined to calculate optical flow by first selecting dominant features, and then tracking them along the video sequences. Motion vectors can be derived from matching features between two frames.

3.2.1 Selecting Features to Track

To avoid ambiguity and aperture problem, intuitively, a good feature should have texturedness and corner, which are determined by eigenvalues of expression 3.2 in Shi[24]'s conclusion. Higher eigenvalue means better feature.

$$\begin{bmatrix} \sum_{neighbour}(\frac{\partial F}{\partial x})^2 & \sum_{neighbour}(\frac{\partial^2 F}{\partial x \partial y}) \\ \sum_{neighbour}(\frac{\partial^2 F}{\partial x \partial y}) & \sum_{neighbour}(\frac{\partial F}{\partial y})^2 \end{bmatrix} \tag{3.2}$$

where F is the intensity of the pixel in the frame, and ∂x, ∂y are the horizontal and vertical displacements of the center of the window containing the neighbour. In OpenCV, $cvGoodFeaturesToTrack()$ is an initial step for optical flow method. But the function is only pixel accurate, $cvFindCornerSubPix()$ afterwards can locate the features up to sub-pixel accuracy by iteration.

3.2.2 Pyramidal Lucas-Kanade Feature Tracker

As one of the widely used differential methods to calculate optical flow, Lucas-Kanade is based on three assumptions as additional conditions: 1) Intensity constancy. Intensity of a small region of the frame remains the same although its location may change. Otherwise, it is not possible to track the features selected by $cvGoodFeaturesToTrack()$. 2) Temporal persistence. The image motion of a surface patch changes gradually over time. 3) Spatial coherence. Neighbouring points in the frame have similar motion if they belong to the same surface. In OpenCV, classical Lucas-Kanade method is implemented as function $cvCalcOpticalFlowLK()$. But it suffers from a dilemma between accuracy and robustness. On one hand, the integration window size $[w_x, w_y]$ in Equation 3.1 should be small enough to make details dominant in the similarity function. On the other hand, the maximum motion vector estimated is limited by the size of the integration window: $\delta_x < w_x$ and $\delta_y < w_y$. Then it is preferable to set the window size $[w_x, w_y]$ large for robustness. As a solution to the dilemma, a pyramidal implementation of classical Lucas-Kanade algorithm is proposed [2]. The main advantage of a pyramidal implementation is keeping relatively small integration window to ensure accuracy while able to estimate large motion vector.

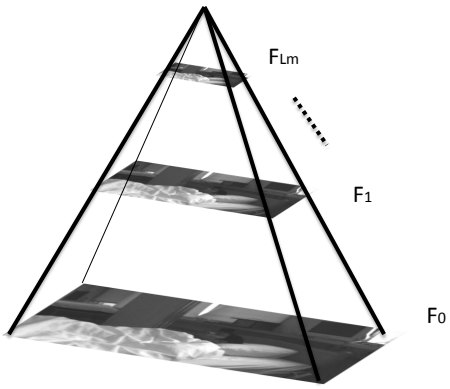


Figure 3.3: Pyramid structure from 0^{th} level to L_m^{th} level. 0^{th} level is the original frame F_0, with the highest resolution. The size of frame in level L_m is $1/(2^{L_m})$ of the original one.

The pyramidal construction of a frame F is recursive. Let F_0 be the 0^{th} level frame(raw frame), with the highest resolution. Each higher level is sub-sampled from the lower frame, for instance, F_1 from F_0, F_2 from F_1, until the highest level F_{L_m}, as illustrated in Figure 3.3. The obvious advantage of a pyramidal implementation is that residue motion vector in each level can be kept very small while the overall motion vector is large. A pyramid depth of $L_m = 3$ can lead to a gain of 15 in overall pixel displacement. $cvCalcOpticalFlowPyrLK()$ is the available implementation in OpenCV .

3.3 RANdom SAmple Consensus(RANSAC)

Motion vectors estimated by Lucas-Kande method can not be all accurate, since it is possible that few features are wrongly tracked. However, those outliers in motion vectors can be filtered out based on majority of inliers. It is assumed that the model determined by certain amount of inliers can distinguish all other inliers from outliers. So as a randomized algorithm, the basic steps of RANSAC are:

- Randomly choose a subset s of available motion vectors S.
- Evaluate a model M from the chosen subset S.
- Evaluate all motion vectors in $S - s$. If the percentage of fitting vectors is large than the threshold t, the model M is returned, and all fitting vectors are valid.
- Otherwise, randomly choose another subset.

By running RANSAC to motion vectors obtained from estimation algorithm, those outliers can be deleted, and the estimation of the global motion is more accurate and robust.

3.4 Analysis of Motion Robustness

In order to analyse the motion robustness of all available algorithms, several test videos are recorded. The reference of respiratory signal is obtained from a contact-based method using the chest strap.

3.4.1 Experiment Set-up

The set-up is illustrated in Figure 3.4. The uEye camera with a fixed 20 frame rate and monochrome is used. The video is stored in format of PFSPD(Philips File Standard for Pictorial Data). In the meanwhile, a chest strap with Piezo transducer is attached to the subject. Respiration signal acquired by the databox from the chest strap is a reference. The synchronization between video and reference is guaranteed by TCP/IP mechanism. The frequency of the respiration is less than 1 Hz, while the sampling frequency of the databox is 1KHz. Thus, the corresponding respiration signal is sub-sampled from the databox by TCP/IP connection in the computer for every frame.

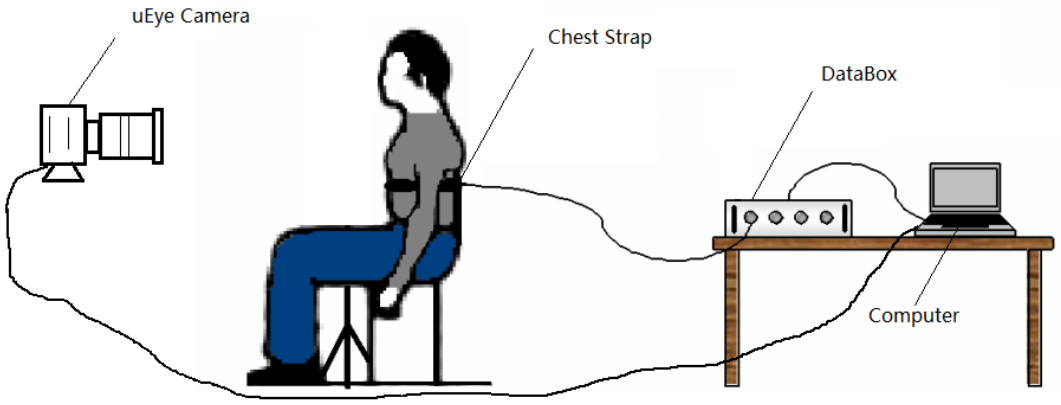


Figure 3.4: Experiment set-up: uEye camera, chest strap with Piezo transducer, DataBox to acquire respiration signal, computer

There are 12 test videos in total, snapshots in Figure 3.5, which can be categorized by four properties: subject, posture of the subject, camera status and orientation of the camera. All the videos are listed in Table 3.1. The label of each video represents the four properties. For example, S1LbHF represents the video of Subject 1 Lying back, with a Hand-held camera in Front. The subject in the video is asked to follow a fixed pattern of various respiration phases: normal, fast, deep, apnea and talking.

Table 3.1: Lists of 12 test videos, categorized by 4 properties. The number of frames and label are also presented.

Video No.	Subject	Posture	Camera-status	Orientation	No. of frames	Label
1	S1	Lying back	Hand-held	Front	2000	S1LbHF
2	S1	Lying back	Hand-held	Side	2000	S1LbHS
3	S1	Lying down	Hand-held	Front	1600	S1LdHF
4	S1	Lying down	Hand-held	Side	2000	S1LdHS
5	S1	Sitting	Hand-held	Front	1600	S1SHF
6	S1	Sitting	Hand-held	Side	1700	S1SHS
7	S2	Lying back	Hand-held	Front	2000	S2LbHF
8	S2	Lying back	Hand-held	Side	2000	S2LbHS
9	S2	Lying down	Hand-held	Front	1600	S2LdHF
10	S2	Lying down	Hand-held	Side	2000	S2LdHS
11	S2	Sitting	Hand-held	Front	1600	S2SHF
12	S2	Sitting	Hand-held	Side	1700	S2SHS

(a) Snapshot of S1LbHF (b) Snapshot of S2SHS

Figure 3.5: Snapshots of two test vides S1LbHF and S2SHS listed in Table 3.1

3.4.2 Quantification

In order to evaluate the motion robustness, two types of quantification metrics can be defined. On one hand, the motion robustness can be evaluated directly by metrics related with the frame, such as pixel-based Peak Signal to Noise Ratio (PSNR) or similarity-based correlation. On the other hand, the accuracy of respiration measurement, such as respiration rate, can be evaluated, which is the ultimate purpose.

Frame-based Metrics

In this section, the performance of the stabilization algorithm is compared based on PSNR and correlation among all 12 test videos recorded in section 3.4.1.
Peak Signal to Noise Ratio (PSNR) , is defined as:

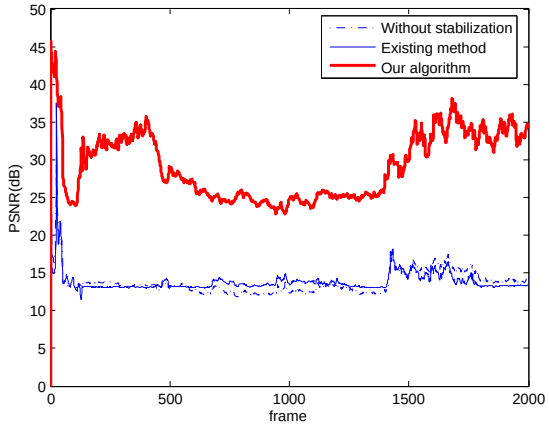
$$PSNR = 10\log_{10}(\frac{Max_{F_I}^2}{MSE}) \tag{3.3}$$

where Max_{F_I} is the maximum possible pixel value of the image, here is 255. And MSE (Mean Square Error) for two images I and J with the same size $m * n$ is defined as

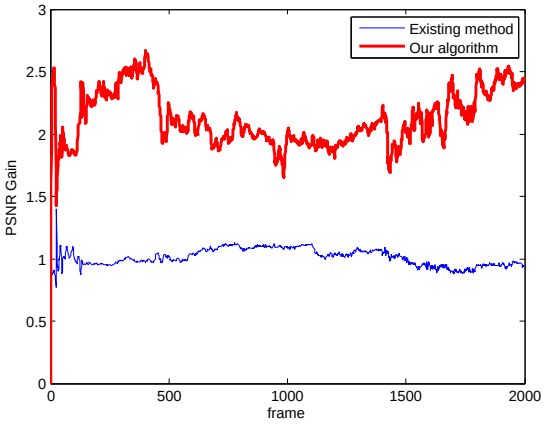
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - J(i,j)]^2 \tag{3.4}$$

If compensated image is identical to the reference, PSNR should be infinite since the MSE will be zero. Compared with other pixel-based criteria, PSNR is best choice for image distortion caused by translation, rotation and scaling. Furthermore, since we have used interpolation in the compensation, it is also important to measure the quality of the reconstruction. For each frame in a video, stabilized frame is compared with the first frame as reference. Thus, in Figure 3.6(a), our algorithm is compared with existing algorithm for test video S1LbHF. From the PSNR gain $= PSNR_{test}/PSNR_{withoutstabilization}$ (Figure 3.6(b)), we can see that our algorithm improves PSNR by the gain of 2 in average.
Correlation between two images I and J with the same size $m * n$ is defined as

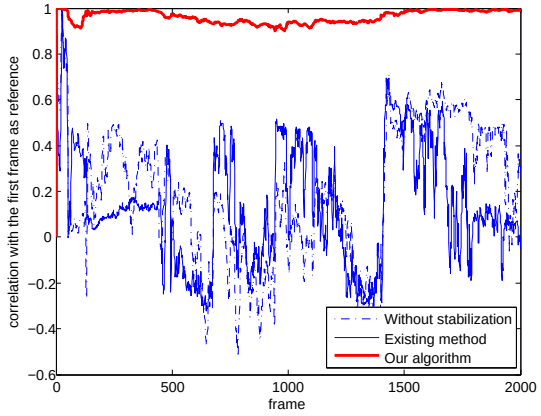
$$corr = \frac{\sum_m\sum_n(I_{mn} - \bar{I})(J_{mn} - \bar{J})}{\sqrt{\sum_m\sum_n(I_{mn} - \bar{I})^2\sum_m\sum_n(J_{mn} - \bar{J})^2}} \tag{3.5}$$



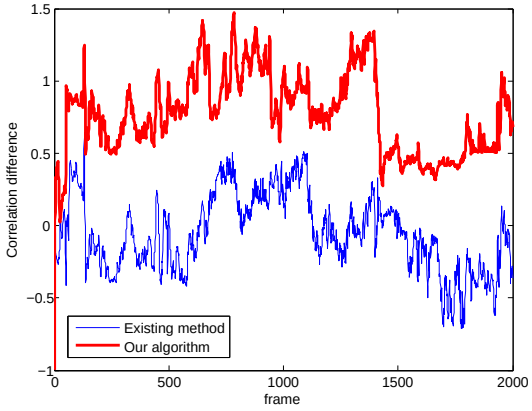
(a) PSNR



(b) PSNR gain derived from Figure 3.6(a)



(c) Correlation



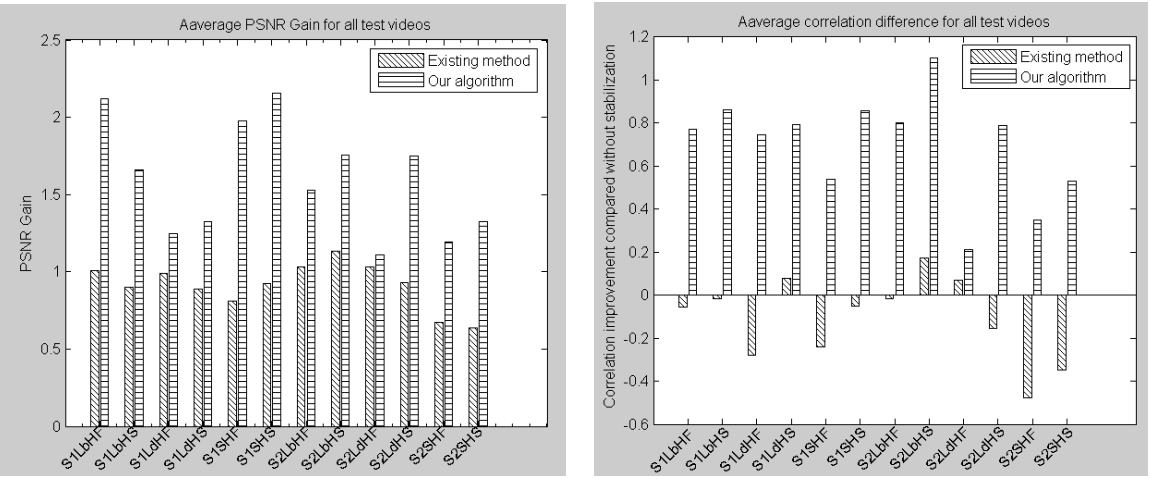
(d) Correlation difference derived from Figure 3.6(c)

Figure 3.6: Performance of stabilization algorithm of test video S1LbHF. The reference frame is the first frame. The stabilized frame is compared with the reference.

where $\bar{I}$ and $\bar{J}$ is the mean value of image I and J.
The benchmark based on correlation for test video S1LbHF is illustrated in Figure 3.6(c). And by comparing the correlation difference $= correlation_{test} - correlation_{withoutstabilization}$ (Figure 3.6(d)), our algorithm surpasses the existing one by 0.8241 in average. For all test videos, mean values of their PSNR gain and correlation difference are compared in Figure 3.7. As the results show, the stabilization algorithm compared with existing one improves PSNR by the gain of 1.77 in average for all test videos, and by the difference of 0.8063 in correlation. But it should be noted that there is still residue motion since PSNR of our stabilization algorithm is not infinite in Figure(3.6(a)).

Respiration-based Metrics

The frame-based metrics are intermediate evaluations considering our goal to measure respiration rate. Therefore, certain metrics based on respiration rate and curve are necessary. For the respiration rate, both the value and timing are important. However, this metric also



(a) Comparison of average PSNR

(b) Comparison of average correlation difference

Figure 3.7: Comparison of average PSNR and correlation difference among all 12 test videos (Table 3.1). The result of each video is derived as the same in Figure 3.6

depends on the performance of peak detection algorithm. A more strict and independent metric is to compare respiration curve directly, such as the correlation with the respiration curve of reference.

Respiration Rate can be calculated by post processing from raw respiration signal using Equation 2.2. Each respiration rate value is coupled with its time stamp in frame. Both the value and timing are important in respiration monitoring, since lagging can contribute to inaccuracy of the respiration rate. Therefore, instead of simply comparing respiration rate value one by one using metrics like mean square error(MSE), the area covered by the respiration rate curve is evaluated. The performance of this metric is influenced by peak detection algorithm, since the respiration rate is determined by peaks.

Suppose the respiration rate measured by a algorithm is $[rr, t]$, where the value of respiration rate $rr = rr_1, rr_2, ..., rr_n$, and corresponding time stamp $t = t_1, t_2, ..., t_n$. A respiration rate curve can be derived as a stair function

$$RR(t) = \begin{cases} 0 & : \quad t < t_1 \\ rr_i & : \quad t_i \leq t \leq t_{i+1}, i \in [1, n-1] \\ rr_n & : \quad t > t_n \end{cases}$$

Then the sum of absolute difference in area (Figure 3.8) covered by the two respiration rate curves $RR_1(t)$ and $RR_2(t)$, referred as SADA is

$$SADA = \int_0^t |RR_1(t) - RR_2(t)|dt \tag{3.6}$$

Metrics of **Respiration Curve**, such as correlation of respiration curve, are more direct and independent compared with the area covered by the respiration rate curve. The correlation is calculated by Equation 3.7. If two respiration curves $X = x_1, x_2, ..., x_n$ and $Y = y_1, y_2, ..., y_n$, the correlation between these two curves is

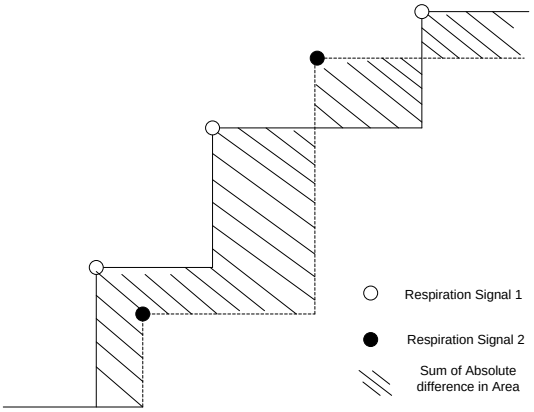


Figure 3.8: Sum of absolute difference in area covered by respiration rate curve.

$$corr = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{X})^2 \sum_{i=1}^{n}(y_i - \bar{Y})^2}} \tag{3.7}$$

where$\bar{X}$ and$\bar{Y}$ are the mean values of two curves. Considering we want to compare the similarity in peaks, correlation between two raw respiration curves will suffer from their inherent DC signals. Therefore, a window based correlation metric is applied. The final correlation is the average of all correlations in each window. Result based on respiration curve is presented in Figure 4.11.

3.5 Conclusion

In this chapter, we first introduced the general concept of global motion estimation and motion model. We have implemented the most suitable algorithm to our scenario, feature-based Pyramid Lucas-Kanade with affine model. The motion robustness is improved according to the benchmark. But it is not always accurate for respiration measurement, since there is still residue motion in the stabilized video.

Chapter 4

Post-Stabilization Algorithm

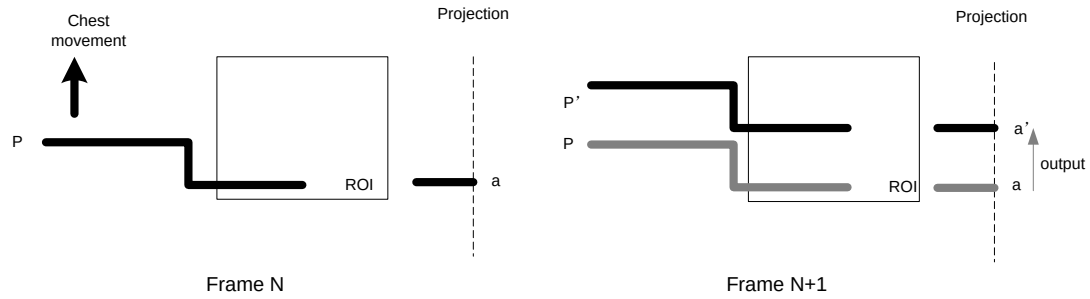
In the last chapter, a feature-based motion estimation algorithm with affine model is implemented to compensate a global motion. However, there is still residue motion in the stabilized video from the benchmark result in Figure 3.6(a), since PSNR is not infinite. In this chapter, the impact of the residue motion on the respiration measurement algorithm ProCor is analysed, and a novel post stabilization algorithm is proposed to further eliminate the error induced by the residue motion and the accuracy of respiration measurement is improved according to the results.

4.1 Residue Motion's Impact on Respiration Measurement

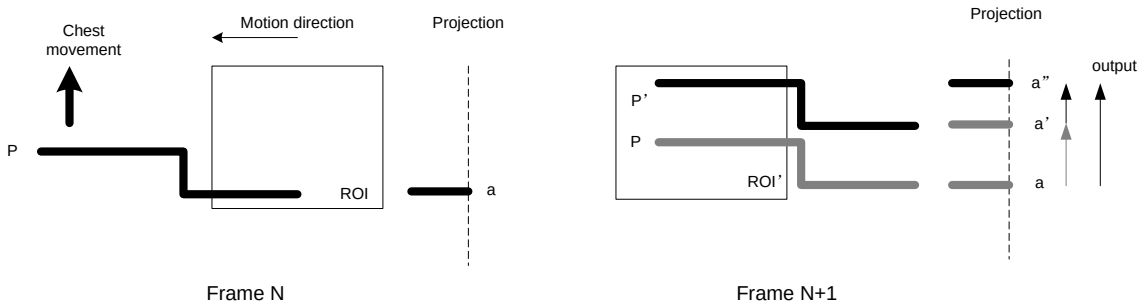
As illustrated in Figure 2.3, the final respiration curve is integration of a shift vector ($\overrightarrow{SV}$) between consecutive frames over time, and the shift is calculated by cross correlation of projection curves in vertical axis of two RespROIs in the neighbouring frames. In this section, we can see how horizontal and vertical global motion are different in affecting the shift vector.

4.1.1 Horizontal Residue Motion

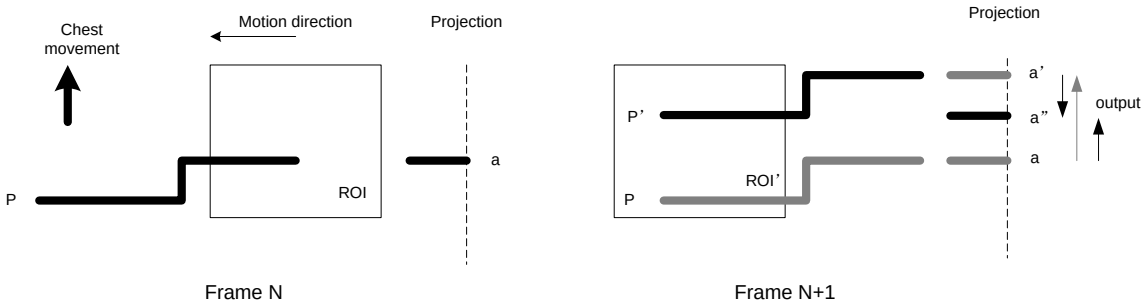
We first assume that the frame is only a stair line P, representing the chest, to simplify the situation, as depicted in Figure 4.1. In Figure 4.1(a), no residue motion exists. The output of ProCor $\overrightarrow{SV} = \vec{aa'}$ is the exact shift vector of the respiration signal. However, if a horizontal residue motion exists(Figure 4.1(b)), the output vector $\vec{aa''}$ is no longer accurate, since it also contains the shift vector caused by the residue motion $\overrightarrow{SV}_{motion} = \vec{a'a''}$. Because of the residue motion, ROI moves, then the projection of ROI in vertical axis is also impacted by the residue motion, contributing to $\overrightarrow{SV}_{motion}$.
In Figure 4.1(b), we can see that with residue motion, the relation between the output of ProCor $\overrightarrow{SV}_{output}$ and the actual respiration signal $\overrightarrow{SV}_{resp}$ is $|\overrightarrow{SV}_{output}| = |\vec{aa''}| = |\vec{aa'}| + |\vec{a'a''}| = |\overrightarrow{SV}_{resp}| + |\overrightarrow{SV}_{motion}|$, but this is not always the case. The relation depends on the direction of residue motion and the content of the frames as well. Comparing Figure 4.1(b) with Figure 4.1(c), the same horizontal motion vector can yield two different shift vectors with opposite directions. Therefore, the outpofuts ProCor are also different. In Figure 4.1(c), $|\overrightarrow{SV}_{output}| = |\vec{aa''}| = |\vec{aa'}| - |\vec{a'a''}| = |\overrightarrow{SV}_{resp}| - |\overrightarrow{SV}_{motion}|$. It should be noted that if $\overrightarrow{SV}_{motion}$



(a) The output shift vector $a\vec{a}'$ is exactly the respiration signal caused by chest movement if no residue motion exists. ProCor's measurement is accurate without residue motion.



(b) The output of ProCor is no longer accurate with horizontal residue motion. The output shift vector $a\vec{a}''$ not only contains the actual respiration signal $a\vec{a}'$caused by chest movement, but also the shift vector $a'\vec{a}''$ caused by the movement of ROI, which is an impact from residue motion. , and $|a\vec{a}''| = |a\vec{a}'| + |a'\vec{a}''|$



(c) Same horizontal residue motion as in Figure 4.1(b) , but the content of the frame is inverse. The output of ProCor $a\vec{a}''$ also changes, because the shift vector $a'\vec{a}''$ has the opposite direction compared with $a'\vec{a}''$ in Figure 4.1(b), and $|a\vec{a}''| = |a\vec{a}'| - |a'\vec{a}''|$

Figure 4.1: Impact of the horizontal residue motion on respiration algorithm

is large enough, the output $\overrightarrow{SV}_{output}$ can have totally opposite direction of actual respiration signal $\overrightarrow{SV}_{resp}$.
In conclusion, the output of ProCor is the sum of two shift vectors of respiration signal and residue motion. The direction of the shift vector caused by residue motion is not only determined by the direction of residue motion, but also the content of ROI in the frame.

4.1.2 Vertical Residue Motion

For vertical residue motion, the relation between output shift vector $\overrightarrow{SV}_{output}$ and residue motion vector is illustrated in Figure 4.2. Without residue motion, we know that the measurement is accurate (Figure 4.2(a)). The analysis becomes a bit more complex with residue motion, as shown in Figure 4.2(b) and 4.2(c). But the output of ProCor $\overrightarrow{SV}_{output}$ is always equal to the sum of $\overrightarrow{SV}_{resp}$ and $\overrightarrow{SV}_{motion}$. And for vertical motion, $\overrightarrow{SV}_{motion}$ is the reversed motion vector, and not influenced by the content of ROI in the image. This is the main difference compared with horizontal residue motion.

4.1.3 Fundamental Limitation due to ProCor

It should be noted that the above analysis does not consider certain cases when correlation in ProCor itself is wrong. As shown in Figure 4.3, when there are missing or appearing peaks of projection, correlation in ProCor will definitely return a wrong shift vector. This situation is out of scope of further investigation. Moreover, if the image in ROI is not sensitive to residue motion, for instance, a full black ROI, the respiration measurement can not work as well. When selecting the ROI manually, it should be aware to avoid these cases.

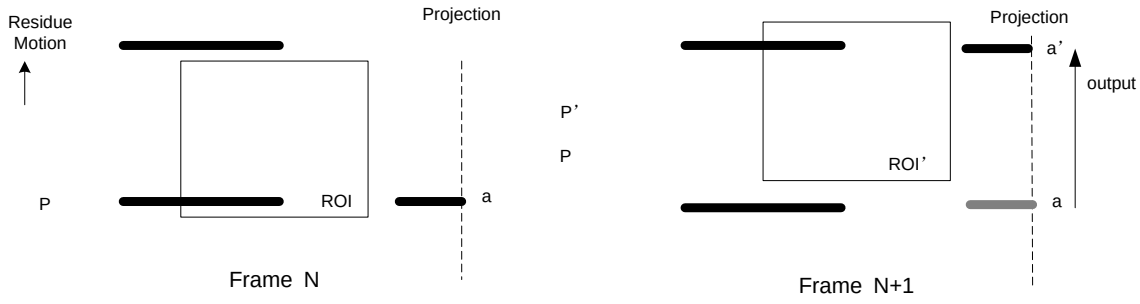
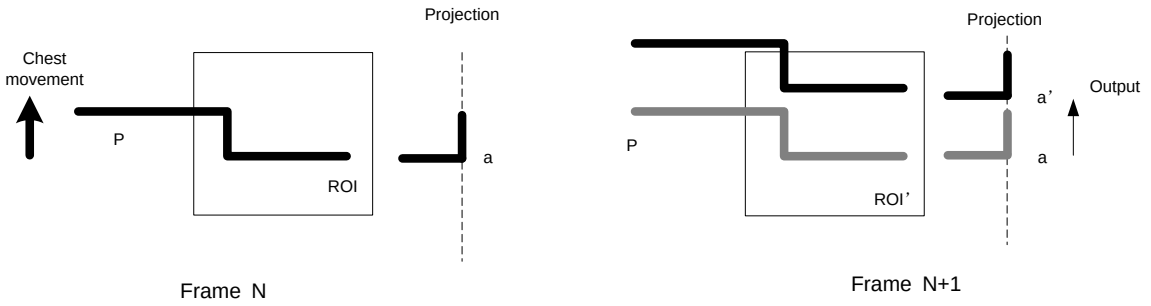


Figure 4.3: Missing and appearing peaks: shift vector from ProCor is not accurate due to mismatched peaks. This is a limitation of ProCor itself.

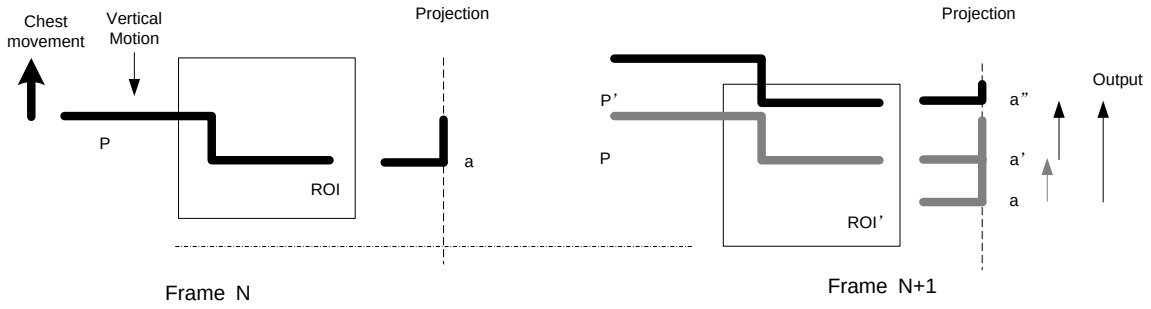
4.2 Improvement based on NoRespROI

In this section, we propose a novel post stabilization algorithm (Figure 4.4) based on the property of residue motion's impact on the respiration measurement algorithm. The idea is using another ROI outside the chest area, called NoRespROI, to compensate the motion-polluted respiration curve derived from RespROI, based on Equation 4.1.

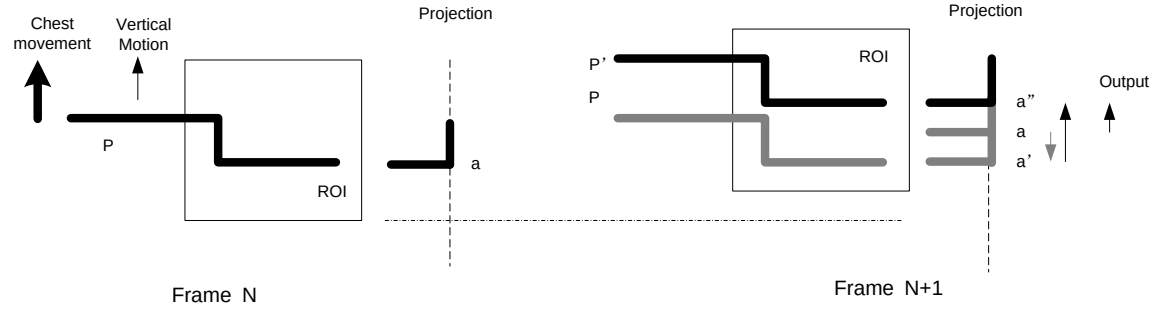
$$\begin{aligned} \overrightarrow{SV}_{compensated} &= \overrightarrow{SV}_{RespROI} - \overrightarrow{SV}_{NoRespROI} \\ &= (\overrightarrow{SV}_{Resp} + \overrightarrow{SV}_{MotionROI}) - \overrightarrow{SV}_{MotionNoROI} \end{aligned} \tag{4.1}$$



(a) The output shift vector is calculated by correlation between projections of ROI of consecutive frames. When there is no residue motion, the respiration measurement is accurate.



(b) The output of PorCor $a\vec{a}'' = a\vec{a}' + a'\vec{a}''$, where $a\vec{a}'$ is induced by the residue motion, and $a'\vec{a}''$ is the real respiration signal. The direction of shift vector caused by residue motion $a\vec{a}'$ is opposite to the direction of the motion.



(c) The direction of shift vector $a\vec{a}'$ is still in the opposite direction of the residue motion, which is not influenced by the content of the frame.

Figure 4.2: Impact of the vertical residue motion on respiration algorithm

In RespROI, both local chest movement and residue motion exist. But in NoRespROI, outside of the chest area, only contains the global residue motion. The residue motion is the same in both ROI, but the shift vector caused by the same residue motion of RespROI $\overrightarrow{SV}_{MotionROI}$ and NoRespROI $\overrightarrow{SV}_{MotionNoROI}$ are different, since they are also influenced by the content of the images in both ROIs. Ideally, if images in RespROI and NoRespROI are the same, then $\overrightarrow{SV}_{MotionNoROI} = \overrightarrow{SV}_{MotionROI}$, the compensated output is exactly the respiration signal accroding to Equation 4.1. Unfortunately, the assumption is not valid in practice. Therefore, the performance of PSA mainly depends on how well NoRespROI is chosen.

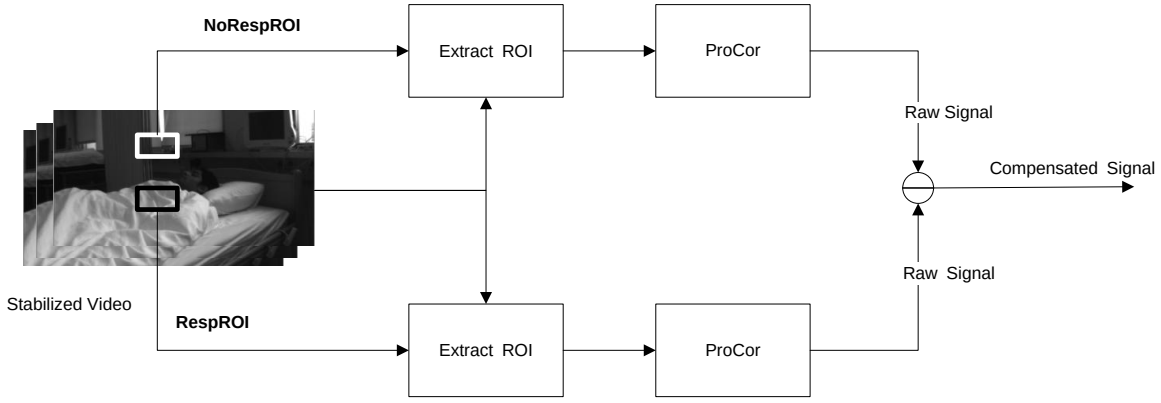


Figure 4.4: Overview of post stabilization algorithm(PSA). Motion-polluted signal extracted from RespROI after stabilization is compensated by signal extracted from NoRespROI.

4.3 Selection of NoRespROI

The principle to select NoRespROI is:

PSA can fully eliminate the residue motion's impact on respiration measurement as long as the shift vectors caused by residue motion of RespROI and NoRespROI are same.

Before we derive certain criteria, one assumption has to be made about ROI. Suppose $MaxResidueMotion$ is the maximum residue motion in pixels of translation, then ROI is assumed to be limited within a rectangle, noted as ROI_{extend} (Figure 4.5), whose width and height are both $MaxResidueMotion$ larger than those of ROI in every frame of stabilized video. Instead of looking at original ROI, we should focus on ROI_{extend}, because ROI_{extend} contains all possible impacts from residue motion.

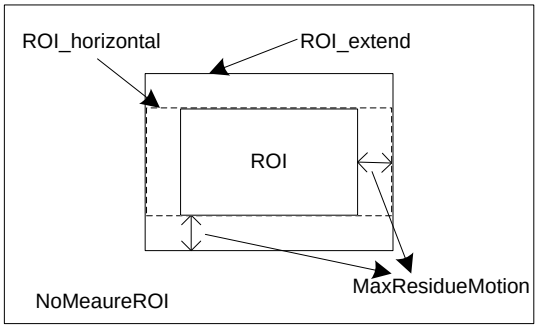


Figure 4.5: Relation between ROI, ROI_{extend} and $ROI_{horizontal}$

Therefore, certain criteria can be derived to find NoRespROI in the frame.

1. Variance of images in $NoRespROI_{extend}$ should be equal or larger than the variance of images in $RespROI_{extend}$. It makes sure that the sensitivity of both ROIs to respiration algorithm ProCor are similar. So residue motion in vertical direction can be fully compensated according to the analysis in section 4.1.2.

2. Correlation between *projection matrix* of $RespROI_{horizontal}$ and $NoRespROI_{horizontal}$ should be maximum, where $RespROI_{horizontal}$ has the same height as $RespROI$ but is $2 * MaxResidueMotion$ wider, as illustrated in Figure 4.5. The *projection matrix*, in which there are 2*MaxResidueMotion+1 columns and ROIHeight rows, can be derived from $ROI_{horizontal}$. Each column is the projection curve of one possible ROI among $ROI_{horizontal}$ (Figure 4.6).

3. $NoRespROI_{extend}$ can not contain any pixels influenced by the local respiration motion. In principle, the chest and abdomen of the subject should be segmented and avoided. We use another ROI called NoMeasureROI to protect area with respiration signal (Figure 4.5). The width and height of NoMeasureROI are both two times larger than RespROI.

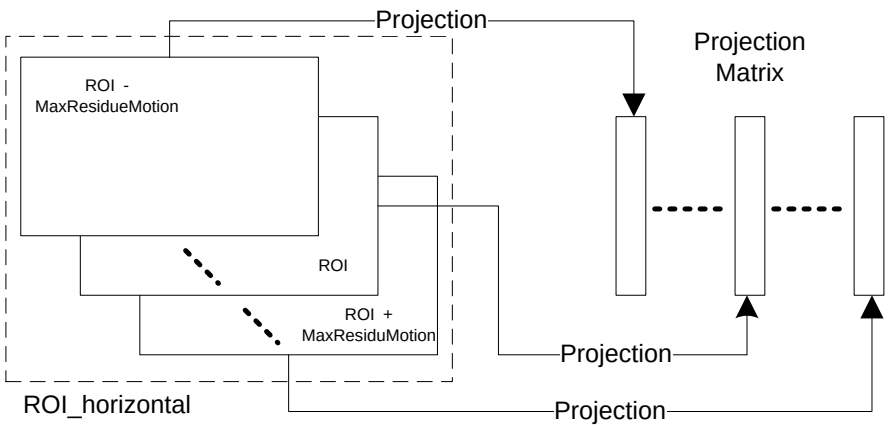


Figure 4.6: Construction of projection matrix from $ROI_{horizontal}$

Pseudo code of PSA based on above three criteria to automatically select NoRespROI from manually selected RespROI:

Algorithm $AutoSelectROI(frame, RespROI, MaxResidueMotion)$
1. Calculate $RespROI_{extend}$ from $RespROI$ and$MaxResidueMotion$
2. Calculate projection matrix of $RespROI_{extend}$ as $RespROIProj$
3. Calculate variance of $RespROI_{extend}$ as $RespROIVar$
4. $NoMeasureROI$ is calculated from $RespROI$
5. **for** $i \leftarrow MaxResidueMotion : FrameWidth - MaxResidueMotion$
6. **for** $j \leftarrow MaxResidueMotion : FrameHeight - MaxResidueMotion$
7. Choose one $NoRespROI$ at position (i, j)
8. **if** $NoRespROI$ is outside $NoMeasureROI$
9. Calculate variance of $NoRespROI_{extend}$ as $NoRespROIVar$
10. Calculate projection matrix of $NoRespROI_{extend}$ as $NoRespROIProj$
11. $correlation$ between $NoRespROIProj$ and $RespROIProj$
12. **if** $correlation > MaxCor$ && $NoRespROIVar > RespROIVar$
13. $MaxCor = correlation$
14. $OptNoRespROI = NoRespROI$
15. **return** $OptNoRespROI$

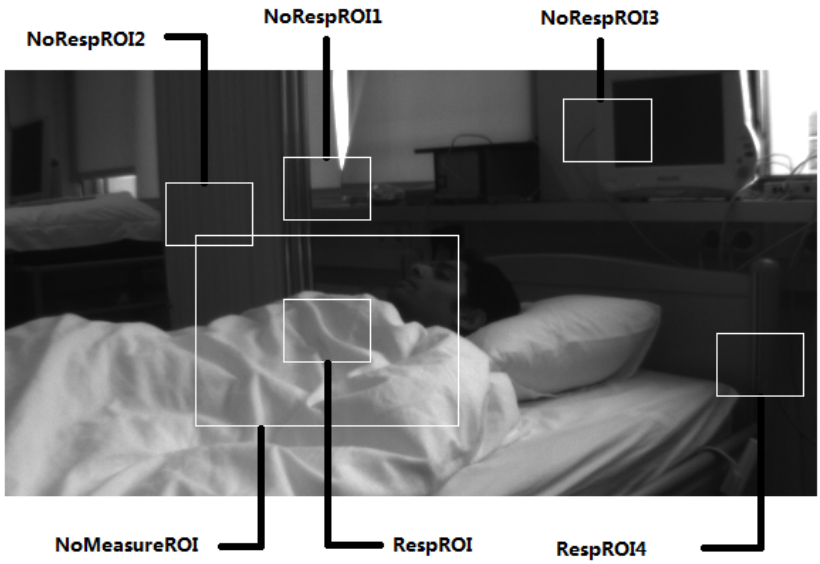


Figure 4.7: Comparison of PSA's performance between four different NoRespROIs: NoResp1 fulfils criteria 1 and 3, NoResp2 does not fulfil any criterion, NoResp3 fulfils criterion 3, and NoResp4 fulfil all criteria. Comparison of their performances are depicted in Figure 4.8

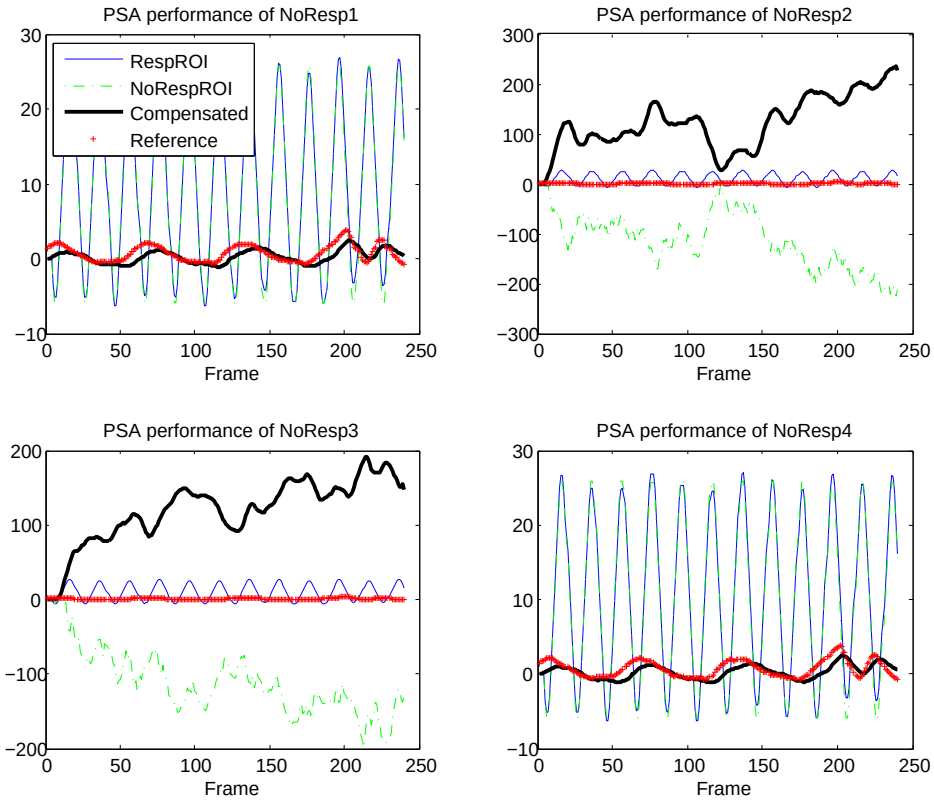
The above algorithm is illustrated by comparing four different NoRespROIs (Figure 4.7) with simulated residue motion. Original video is recorded in experiment mentioned in subsection 3.4.1 with still camera. Simulated horizontal and vertical residue motions in triangular wave are added into original video, with a period of 20 frames and amplitude of $[-5, 5]$ pixels.

The variance and correlation with RespROI of four NoRespROI are listed in Table 4.1. Only NoRespROI4 fulfils all three criteria.

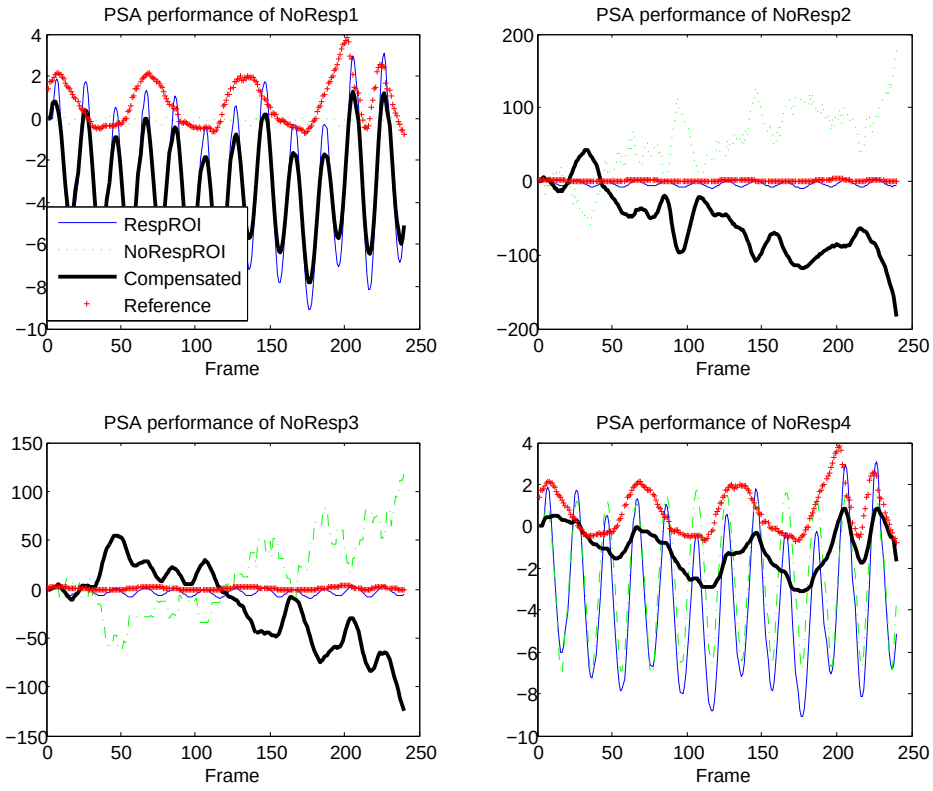
Table 4.1: Four NoRespROIs: their variances and correlation with RespROI. Noted: variance of RespROI is 18.8671.

NoRespROI	Variance	Correlation	Criterion 1	Criterion2	Criterion 3
1	30.0499	-0.1445	√	×	√
2	1.6327	-0.1904	×	×	×
3	13.8242	0.0449	×	×	√
4	19.8242	0.2389	√	√	√

The performances of PSA with above four NoRespROIs are compared in Figure 4.8. From the above results, PSA with NoResp2 and NoResp3 can not compensate any residue motion, PSA with NoResp1 can only compensate vertical motion, and PSA with NoResp4 can fully eliminate residue motion. Therefore, we can conclude that PSA can compensate both horizontal and vertical residue motion if NoRespROI satisfies all criteria 1, 2 and 3.



(a) Vertical simulated motion



(a) Horizontal simulated motion

Figure 4.8: Performance of PSA on simulated video with four different NoRespROIs in Figure 4.7

4.4 Results

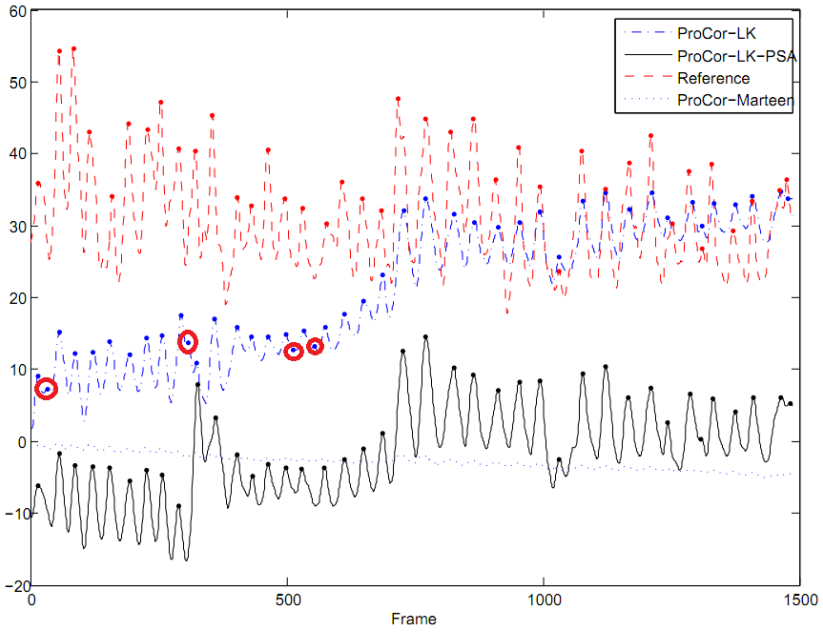
In this section, the performance of PSA is evaluated by practical videos listed in Table 3.1. We try to compare the performance of original respiration measurement algorithm ProCor, existing compensation algorithm ProCor-Marteen, our compensation algorithm ProCor-LK and post stabilization algorithm ProCor-LK-PSA using respiration-based metrics (section 3.4.2).

In Figure 4.9(a), the respiration curve and corresponding respiration rate derived from video S2SHS are depicted. By comparing detected peaks in reference with test algorithms, four wrong peaks in ProCor-LK is deleted by PSA, and the corresponding respiration rate is more accurate (Figure 4.9(b)). Actually, there are two types of inaccurate measurement: 1) missing peaks(MP), if peak in the reference is not detected in test algorithm. 2) wrong peaks(WP), if peak in the test algorithm is not a peak in the reference. All statistics of missing peaks and wrong peaks for all test videos are listed in table 4.2.

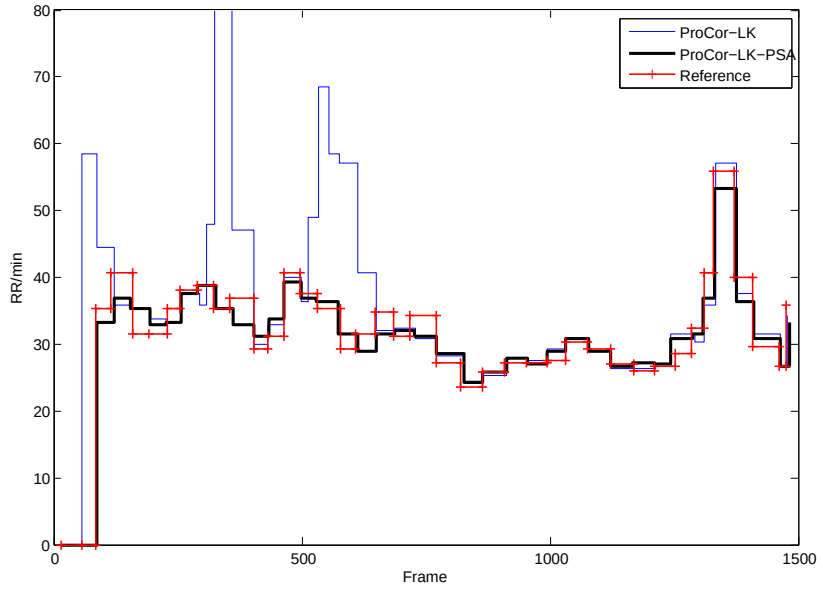
Table 4.2: Statistics of missing peaks (mp) and wrong peaks(wp) in 12 test videos

Video No.	Algorithm	MP	WP	MP Ratio	WP Ratio
S1LbHF	ProCor-LK	4	0	017	0
	ProCor-LK-PSA	3	3	0.13	0.13
S1LbHS	ProCor-LK	5	3	0.16	0.09
	ProCor-LK-PSA	3	2	0.09	0.06
S1LdHF	ProCor-LK	12	3	0.38	0.09
	ProCor-LK-PSA	7	2	0.21	0.06
S1LdHS	ProCor-LK	10	3	0.29	0.09
	ProCor-LK-PSA	6	0	0.17	0
S1SHF	ProCor-LK	5	2	0.14	0.06
	ProCor-LK-PSA	3	1	0.09	0.03
S1SHS	ProCor-LK	4	2	0.11	0.06
	ProCor-LK-PSA	2	1	0.6	0.03
S2LbHF	ProCor-LK	1	0	0.04	0
	ProCor-LK-PSA	1	0	0.04	0
S2LbHS	ProCor-LK	4	0	0.11	0
	ProCor-LK-PSA	3	0	0.08	0
S2LdHF	ProCor-LK	6	0	0.16	0
	ProCor-LK-PSA	2	0	0.05	0
S2LdHS	ProCor-LK	7	0	0.17	0
	ProCor-LK-PSA	5	0	0.12	0
S2SHF	ProCor-LK	6	0	0.18	0
	ProCor-LK-PSA	5	0	0.14	0
S2SHS	ProCor-LK	1	9	0.02	0.17
	ProCor-LK-PSA	0	5	0	0.10

We can see that PSA can add missing peaks and delete wrong peaks in most cases. The consequent improvement can be illustrated in Figure 4.10. In Figure 4.10, the sum of absolute difference of the area covered by the respiration rate curve with reference is calculated for all 12 test videos, and the respiration measurement is more accurate for smaller difference. We can see improvements in 11/12 videos, except S1LbHF. In video S1LbHF, although 1 missing point is added, PSA also induces 3 additional wrong points. The wrong points induced by PSA is caused by a sudden drop in the respiration curve. At the same time, a big change in the viewing angle is observed in the stabilized video, which is a limitation of the affine model applied in the stabilization algorithm.



(a) Respiration curves of ProCor-LK, ProCor-LK-PSA, ProCor-Marteen and reference in video S2SHS. Four wrong peaks in ProCor-LK are compensated in ProCor-LK-PSA. ProCor-Marteen is not working in this test video.



(b) Respiration rate curves of ProCor-LK, ProCor-LK-PSA and reference in video S2SHS. Four wrong peaks in the respiration curve of ProCor-LK leads to three inaccurate intervals of respiration rate.

Figure 4.9: Benchmark result of video S2SHS.

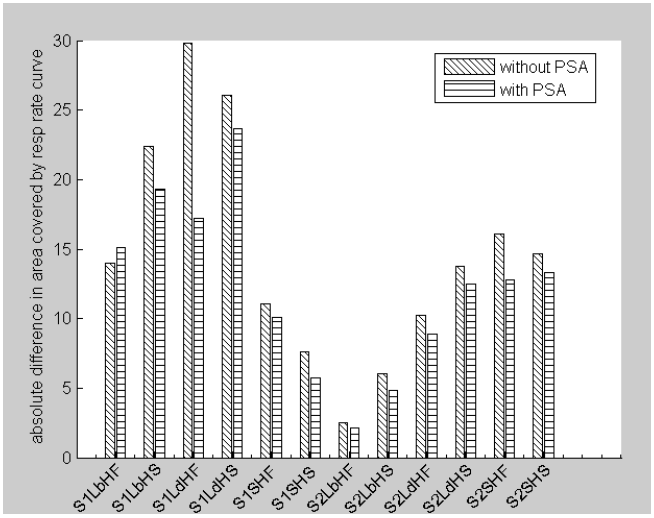


Figure 4.10: The sum pf absolute difference in area covered by respiration rate curve with the reference between with or without PSA are compared. Without PSA is ProCor-LK algorithm, and with PSA is ProCor-LK-PSA.

The improvement can also be verified by correlation metrics defined by Equation 3.7. From Figure 4.11, PSA improves the correlation for 10/12 test videos except S1LbHF and S2LbHF. For video S2LbHF, both ProCor-LK and ProCor-LK-PSA are performing best according to the result in Figure 4.10, and they are the same regarding to missing or wrong peaks. The minor decrease in the performance may be caused by lagging in peaks detected. It is possible that PSA can influence the peak detection algorithm on the timing stamp of peaks.

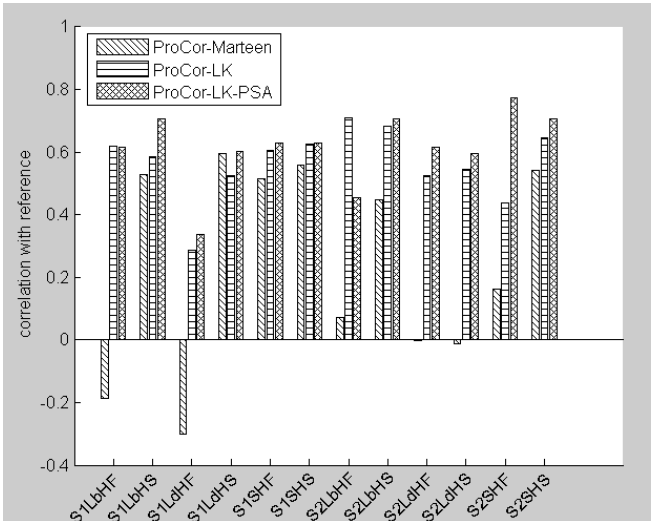


Figure 4.11: The raw respiration curves derived from three algorithms are compared in window-based correlation with reference. The length of the window is 200 frames. ProCor-Marteen is the existing compensation algorithm, ProCor-LK is our stabilization algorithm, and ProCor-LK-PSA includes PSA algorithm after the stabilization

4.5 Conclusion

In this chapter, we have proposed a novel post stabilization algorithm to further eliminate the impact from residue motion by using another ROI called NoRespROI, in which only residue motion exists. The performance of PSA mainly depends on how well NORespROI is chosen. Therefore, an automatic selection of NoRespROI is implemented based on three criteria to guarantee the performance of PSA. From the benchmark result, the accuracy of respiration measurement is indeed improved by PSA.

Chapter 5

Summary

This thesis develops and implements a motion robust algorithm for respiration monitoring using a hand held camera. The original respiration measurement algorithm, referred as ProCor, was developed by Philips. Electronics B.V. previously, however the algorithm is limited to the scenario of stationary camera. In order to improve its motion robustness, a stabilization algorithm in combination of pyramid Lucas-Kanade feature tracker and affine global motion model is implemented. The benchmark shows that the motion robustness is improved, though still suffers from the impact of the residue motion on respiration measurement. Therefore, a post stabilization algorithm is proposed to correct the respiratory signal with non-respiratory motion residue. Another ROI, called NoRespROI is chosen outside the area with respiratory signal to represent the residue motion. The performance is mainly determined by how well NoRespROI is chosen. We have defined certain criteria to select NoRespROI, and the residue motion can be compensated using PSA with those NoRespROIs.

5.1 Future Work

- More complex global motion model. Four parameter affine model can not dealt with subtle but existing perspective transform. Upgrading the motion model may reduce the residue motion after stabilization, and the probability that the residue motion impacts the respiration rate is also reduced.
- Better peak detection algorithm. The accuracy of the respiration rate is also heavily influenced by the peak detection methods. Missing peaks and wrong peaks may be found or discarded for a better peak detection algorithm.
- Real time implementation. In ProCor-LK-PSA algorithm, the motion estimation and compensation algorithm is the most complex and time consuming part. Several improvements are possible, for example, only images in ROI stabilized reduce execution time. Currently, four parameter affine model is used. Additional parameters can improve the accuracy of stabilization, but further increase the computational workload.

Bibliography

[1] J. Biemond, L. Looijenga, DE Boekee, and R. Plompen. A pel-recursive wiener-based displacement estimation algorithm. *Signal Processing*, 13(4):399–412, 1987.

[2] J.Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 2001.

[3] C. Cafforio and F. Rocca. Methods for measuring small displacements of television images. *Information Theory, IEEE Transactions on*, 22(5):573–579, 1976.

[4] P. Csillag and L. Boroczky. New methods to improve the accuracy of the pel-recursive wiener-based motion estimation algorithm. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 3, pages 714–716. IEEE, 1999.

[5] E. De Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5):700–703, 1987.

[6] G. de Haan, P.W.A.C. Biezen, H. Huijgen, and O.A. Ojo. True-motion estimation with 3-d recursive search block matching. *Circuits and Systems for Video Technology, IEEE Transactions on*, 3(5):368–379, 1993.

[7] S. Dingab, X. Zhua, W. Chena, and D. Weia. Derivation of respiratory signal from single-channel ecgs based on source statistics. 2004.

[8] S.N. Efstratiadis and A.K. Katsaggelos. A model-based pel-recursive motion estimation algorithm. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 1973–1976. IEEE, 1990.

[9] V.V. Estrela and N.P. Galatsanos. Spatially-adaptive regularized pel-recursive motion estimation based on cross-validation. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 2, pages 200–203. IEEE, 1998.

[10] R. Vlutters F. de Bruijn, A. Heinrich. Remote respiration analysis, 2001.

[11] C.S. Fuh and P. Maragos. Affine models for image matching and motion detection. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 2409–2412. IEEE, 1991.

[12] J. Geisheimer. Rvsm [radar vital signs monitor]. *Potentials, IEEE*, 17(5):21–24, 1998.

[13] Marek Bartula Gerard de Haan. Apparatus and method for breathing analysis, 2008.

[14] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro. Motion compensated interframe coding for video conferencing. In *Proc. Nat. Telecommun. Conf*, volume 5, pages 1–5, 1981.

[15] H. Li, P. Roivainen, and R. Forchheimer. 3-d motion estimation in model-based facial image coding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(6):545–555, 1993.

[16] R. Li, B. Zeng, and M.L. Liou. A new three-step search algorithm for block motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 4(4):438–442, 1994.

[17] L.K. Liu and E. Feig. A block-based gradient descent search algorithm for block motion estimation in video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(4):419–422, 1996.

[18] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[19] E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *Image Processing, IEEE Transactions on*, 7(5):703–719, 1998.

[20] AN Netravali and JD Robbins. Motion-compensated television coding. i. *AT T Technical Journal*, 58:631–670, 1979.

[21] L.M. Po and W.C. Ma. A novel four-step search algorithm for fast block motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(3):313–317, 1996.

[22] B.S. Reddy and BN Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *Image Processing, IEEE Transactions on*, 5(8):1266–1271, 1996.

[23] V.E. Seferidis and M. Ghanbari. General approach to block-matching motion estimation (journal paper). *Optical Engineering*, 32(07):1464–1474, 1993.

[24] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[25] C. Stiller. Object-based estimation of dense motion fields. *Image Processing, IEEE Transactions on*, 6(2):234–250, 1997.

[26] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ., 1991.

[27] M. van den Helder. Motion robust techniques for camera-based monitoring of respiration. 2011.

[28] D. Walker and KR Rao. Improved pel-recursive motion compensation. *Communications, IEEE Transactions on*, 32(10):1128–1134, 1984.

[29] C. Zhu, X. Lin, and L.P. Chau. Hexagon-based search pattern for fast block motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(5):349–355, 2002.

[30] S. Zhu and K.K. Ma. A new diamond search algorithm for fast block-matching motion estimation. *Image Processing, IEEE Transactions on*, 9(2):287–290, 2000.