Marshall Plan Scholarship Report

Object Tracking in Ultra-Sonic Images

submitted by:

Bernd Altinger, BSc

AUSTRIAN
MARSHALL PLAN FOUNDATION

BGSU®
Bowling Green State University

fhs
Fachhochschule
Salzburg University
of Applied Sciences

Salzburg, July 2013

Contents

List of Figures

List of Tables

Listings

Abbreviations

CAMSHIFT	Continuously Adaptive Mean Shift
CLT	Central Limit Theorem
HOG	Histogram of Oriented Gradients
Hz	Hertz
KLT	Kanade-Lucas-Tomasi
HSV	Hue, Saturation, Value
LAB	L for lightness and a and b stands for the color-opponent dimensions
PCA	Principial Component Analysis
RGB	red, green, blue (colour space)
ROI	region of interest
SIFT	Scale Invariant Feature Transform
SNR	signal to noise ratio
SOAMST	Scale and orientation adaptive mean shift tracking
YUV	luminance, chrominance and intensity (colour space)

1 Chapter 1 Introduction and Motivation

In a currently ongoing research project at the University of Applied Sciences Salzburg, tools for automatic detection of nephrolits inside the kidney from ultra-sonic (US) images are developed . Part of the research questions deal with algorithms that are able to track the motion of objects in a sequence of US images, such as the kidney border, a detected nephrolit or others. During the non-invasive medical treatment of kidney stones with extracorporeal shock wave lithotripsy ESWL, a stone is disintegrated in a series of small steps. For the monitoring of the process and for the verification of "hits" of the shockwave, small movements of certain objects in the imaging area have to be tracked and evaluated. The purpose of the proposed research is on examining the applicability of established template-matching and template-tracking technologies on US images. Here, the smaller signal to noise ratio and the kind of typical artefacts in US images constitute additional challenges. Starting from a quite general framework, optimization by filtering/smoothing and more fault-tolerant pattern recognition techniques shall be examined in order to achieve a stable tracking method.

1.1 Problem statement

The main goal of this thesis is to analyse various object recognition, motion detection and object tracking algorithms and to examine which combination works best for the problem of kidney- and kidney stone-tracking.

1.2 Known approaches

This section gives an insight in object tracking, movement detection algorithms that are already in use for other purposes. Thinking outside the box and looking for similar use cases prevents the "re-invention of the wheel". It also should give an insight on the range of methods available to track objects or to determine motion.

1.2.1 Respiratory-gated radiotherapy

Marco A. Serpa Lopez [14] mentions the respiratory-gated radiotherapy in his thesis for surpressing the tumor treatment during the respiratory cycle of the patient. Due to the damage which ionising radiation causes to cells it is necessary to limit the the radiation dose to the healthy tissue. The respiration can be determined by external sensors like pressure belts, optical reflectors or by internal fiducial markers or fluoroscopy. If a sensor measures the beginning of a respiratory cycle it fires a signal to the radiation device and stops the treatment until the respiratory cycle is completed[14].

To avoid that the ESWL becomes a invasive method by injecting fluids or markers and the defining of fix fiducial markers, without any pre segmentation, for the ultrasound modality is challenging. Adaption of the external sensors seems the most reasonable approach. A simple pressure belt around the chest of the patient would allow the ESWL device to recognize the respiratory cycle and to abort the treatment and to start the reinitialisation of the segmentation algorithm at the end of the breathing cycle. An advantage of this method is the simplicity of the implementation and the theoritcally need for tracking or segmentation algorithms during the respiration. The disadvantages of this method would be, that shallow breathing may stay undetected and all other movements remained undetected.

Another possiblity would be the usage of reflective markers on the skin of the patient or a projected laser grid on the patient to determine motion. Thus, would lead to a need of out-of-body image processing. This line of thought can be considered for future research papers to enable the possiblity to compare various respiratory recognition algorithms and which returns better results.

1.2.2 Tumour Tracking

Marco A. Serpa Lopez [14] suggested also the use of image tracking methods to aid or replace the repiratory-gated therapy. He outlined that most of the tracking algorithms for radiotherapy would use either to try to calculate a correlation between movement measured external to the internal tumor movement, which actual can vary over time and therefore is not suitable, or the usage of internal tumor tracking. To track a tumor internal there are two methods available: with or without markers that are placed or injected into the patient. He also suggested that this implant procedures causing an additional risk of infections and therefore he suggested the usage of tumor tracking algorithms, which work markerless. The only problem of this approach is, that there are not many tumor tracking algorithms yet, which are clinically approved and work without markers[14]. Marco A. Serpa Lopez [14] suggested the usage of the software package PortalTrack, which was developed by the University of Wuerzburg, which was used for used, rated and discussed. The results showed that the software was capable of keeping track of the tumour and therefore this or comparable software are suggested for future research.

2

Chapter 2

Background Information

In this chapter, information is being summarized about the terminology used in the thesis. Additionally some information about the ultrasound imaging system and the problems arising from this modality are being presented.

2.1 Terminology

2.1.1 Image / US-Image

An US-image is an image that has been recorded via an ultrasound device. The images can differ in quality and signal to noise ratio (SNR) depending on the manufacturer and technological level of the device. Ultrasonic images can be characterized by a higher noise level and a worse image quality than images taken e.g. by computer tomography, but also for not exposing patients and personnel to radiation.

2.1.2 Pixel

Definition by George P. Rédei [19]:

> “A picture element in the computer that represents a bit on the monitor screen or in the video memory.” ([19], p. 1501)

Therefore a pixel is the smallest unit in an image and holds a certain value, which can be interpreted as colour, brightness or other representations of various colour spaces. Depending on the maximum number of pixels allowed in an image and the allowed value range of each pixel, images vary in size, resolution, colour, memory usage and possibilities for further processing.

2.1.3 Object

In general, objects are interconnected pixels or regions in an image. The pixels can be 4-connected or 8-connected. The difference between these two is that the 4-connected only counts pixels as a neighbour if the pixels share a side of their edges, whereas 8-connected pixels share either one of their edges or their corners. Image processing is aimed to determine which pixel is more likely to belong to which object. To aid this process, objects features such as described in Section 2.1.4 are needed to describe the desired objects for segmentation, tracking or other image processing algorithms.

An object in an ultrasound image as generally applied in medical circumstances, represents a specific inward part of the human body. The main goal is to track these objects and to distinguish between objects and noise fragments or structures which are not important for tracking measures.

2.1.4 Feature

Features support the segmentation or tracking algorithm in differing between objects and structures. Features are characterized by pixels or region of pixels which represent some numerical attribute of the object. Furthermore, it is important that characteristics are chosen which are individual for each object either in their appearance or in the number of their appearances. The selection of features depends on the image modality and the objects which should be classified. Often, features are combined to optimize image processing. These features help to determine the border or a probabilistic region of some object. Features for object tracking are described in Section 5.1.3.

An example for the image processing process are bright pixels with a certain minimum number to determine whether a bright region is a certain object or some noise in the image.

2.1.5 Image Noise

In general, noise adds some spurious and extraneous information to an image or signal and is an undesirable artifact of the image acquisition process. The characterics of image noise are a random variation of brightness or colour information[9].

Noise often appears as a additive component in images, like Gaussian noise. Let be f an image, g a desired component of the image like a pixel and q a noise component [3]:

$$f = g + q \tag{2.1}$$

Noise can also be multiplicative, like speckles:

$$f = g \cdot q \tag{2.2}$$

Both models can also be converted into one another by either using the exponentiation to convert the multiplicative form into the additive and the logarithm for vice versa [3]. Alan C. Bovik [3] raises the question behind the meaning behind it and replies:

> “The answer is that we are looking for *simple* models that properly describe the behavior of the system.” ([3],p. 397)

There are also some situations where both models fit the noise well. Furthermore, it is important to consider that the noise component q is often related to g. Therefore, the elemination or mitigation of q also affects g. Hence, it is often impossible to remove the noise entirely from the image without affecting the original pixel values severly.

There are various types of noise which can appear in images[4]:

- Gaussian noise: Gaussian noise is a part of almost every signal and most frequently occurs as addivitve Gaussian noise. With the Gaussian it is possible to model thermal or white noise. Let q be the density function of univariate Gaussian noise, μ the mean value, σ^2 the variance and $-\infty < x < \infty$ [4]:

$$p_q(x) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{2.3}$$

  In this case the range where the probality density is nonzero is infinite in the positve and the negative direction. However, in an image the values have to be nonnegative and therefore the noise cannot be strictly Gaussian. In practical use the Gaussian density is a good model for many processes and the range of the Gaussian noise is limited to $\pm 3\sigma$. One of the most significant properties of the Gaussian distribution is the Central Limit Theorem (CLT). According to Alan C. Bovik [4] the CLT:

  > "... states that the distribution of a sum of a large number of independent, small random variables has a Gaussian distribution." ([4],p. 150)

  It it also mentioned, that the variables dot not need a Gaussian distribution themselves, but there are some important conditions which have to be fullfilled [4]:

  - The number of the random variables must be large.
  - The variables have to be independent or approximately independent.
  - Every single variable in the sum has to be small compared to the sum.

  As an example thermal and white noise were mentioned earlier. Thermal noise is a result of vibrations of a large number of electrons. Each vibration is independent and no electron contributes noticeable more than another. As it fullfills all three requirements thermal noise can be modeled as a Gaussian. White noise is defined as a constant value of the power spectral density of the signal for all frequencies [9]. Furthermore, white noise can be modeled as a Gaussian[4] and therefore Gaussian filters are used to reduce the effect of white noise. However, it is impossible to remove Gaussian noise in its entirety [22]. Filtering can improve the quality of the image but only at costs of sharpness[4].

- Heavy-Tailed Noise: Heavy tailed noise can appear if the distribution does not fulfil the requirements for the CLT for the Gaussian distribution. There are cases, where the center of the density can be modeled as a Gaussian but the tails cannot. A tail is defined, where $|x| \to \infty$. Furthermore, a heavy tails is defined as where the density function p(x) for large values of x, approaches 0 slower than the Gaussian function [3]. There are several heavy tailed noise models which are common in image processing [4]:

- Laplacian: The μ value can be estimated best with the median and not with the mean. The Laplacian is not a real type of noise but prediction errors in image compression algorithms can be modeled as Laplacian. This modelling occurs at the difference between consecutive pixels. Let μ be the mean and $\frac{2}{\lambda^2}$ the variance [4]:

$$p_q(x) = \frac{\lambda}{2} \exp(-\lambda |x - \mu|) \tag{2.4}$$

- Alpha-Stable: This model is characterized by a same distribution of appropriately normalized sums of independent and equivalent distributed random variables and the individual random variables. These alpha-stable distributions have some functions, which are most like the form $\exp(-|u|^{\alpha})$, where $0 < \alpha \leq 2$ [3]. The density functions of Alpha-stable distributions cannot be written in closed form except for $\alpha = 1$ (Cauchy) and $\alpha = 2$ (Gaussian). If α tends to zero the Alpha-stable distributions become very heavy tailed [4].

- Generalized Gaussian: Let be μ the mean, A, β and α are constants, where α describes the shape of the density[4]:

$$p_q(x) = A \, \exp\left(-\beta |x - \mu|\right)^{\alpha} \tag{2.5}$$

  If $\alpha = 1$ the shape is like a Laplacian, and if $\alpha = 2$ it corresponds to a Gaussian. For values of α inbetween, the tails are also between the Gaussian and the Laplacian. If $\alpha < 1$ the distributions get even heavier tailed. The parameters A and β are related to α and the standard derivation σ [3]:

$$\beta = \frac{1}{\sigma} \left(\frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})} \right)^{\frac{1}{2}} \tag{2.6}$$

$$A = \frac{\beta\alpha}{2\Gamma(\frac{1}{\alpha})} \tag{2.7}$$

  The advantage of the generalized Gaussian compared to other models is the ability to fit a variety of symmetric noises by a suitable choice of the parameters α, μ and σ.

It is important to know if heavy tailed noise can appear in an image, due to the fact that a mean filter would perform quite well for Gaussian noise but not in heavy-tailed noise. It is recommended to use the median, which performs much better in heavy tailed noise but also performs only slight worse in Gaussian noise compared to a mean.

- Photon counting noise: This noise appears mostly at low-light situations at the image sensor, where the photons are counted[4]. Most image acquisition devices are counting the photons. The photon counting noise often can be modeled as Poisson. Let be a the number of photons at a certain pixel position, k=0,1,2,..., and λ the parameter for the Poisson distribution[4]:

  $$P(a = k) = \frac{e^{-\lambda}\lambda^k}{k!} \tag{2.8}$$

  If λ is large, the Poisson distribution can be reasonably approximated by a Gaussian where the mean and the variance of the Gaussian are equal to λ[4].

  An attribute of the photon counting noise is that it affects brighter regions more than darker ones, because brighter regions have a higher λ[4].

- Salt and Pepper Noise: This type of noise is an example of noise which is very heavy tailed. Salt and pepper noise occurs if only a few pixels are noisy, but these pixels are very noisy. This can appear if the image signal is transmitted over noisy digital links and results in black (pepper) and white (salt) dots on the image. Various order statistic filters can be used to remove salt and pepper noise [3].

- Quantization Noise: If an image is digitialized some quantization has to be done, which can lead to quantization noise[4]. Due to the fact that a continous random variable has to be converted into a discrete random variable, quantization errors are likely to occur. Quantization noise is often modeled as a uniform noise, which is the opposite of heavy-tailed noise described in Section 2.1.5. This results in very light tails. To reduce quantization errors it is recommended to increase the number of quantization levels[3].

- In US-images, the primary cause of image noise and clutter is the composition of the body wall. In detail, fat, skin layer thickness and hydration level are some of the principal causes of ultrasound beam distortion and scattering. Additionally slice thickness, side lobes and reverberation artifacts also contribute to general image clutter. The minimal harmonics are generated from this distorted and scattered energy, which is much weaker than the transmitted energy [22].

Design patterns for noise reduction are beeing examined in [24]. Design and implementation of image filters is being discussed in [18].

3

Chapter 3

Object Segmentation Algorithms

This chapter deals with different approaches of object segmentation algorithms. A good segmentation algorithm should produce regions which are homogneuous and uniform regarding to certain chosen features [26]. Lilly Spirkovska [26] described segmentation in a mathematical way, where I defines all pixels in the image and P() is a uniformity characteristic, which is calculated on a group of interconnected pixels. $\{R_1, R_2, ..., R_n\}$ defines a distribution set of connected image regions [26]:

$$\cup_{l=1}^{n} R_l = I, \, where \, R_l \cap R_m = \oslash \forall \, l \neq m \quad (3.1)$$

$$P(R_l) = True \, \forall l \quad (3.2)$$

$$P(R_l \cup R_m) = False, \, \forall R_l \, adjacent \, to \, R_m \quad (3.3)$$

$$(R_l \supset R_m) \wedge (R_m \neq \oslash) \wedge (P(R_l) = True) \Rightarrow P(R_m) = True \quad (3.4)$$

Bernd Jähne [13]describes object segmentation as follows:

> “Segmentation is the operation at the threshold between *low-level image processing* and *image analysis*. After segmentation, we know which pixel belongs to which object.” [13, page 449]

Due to Jaehne [13] object segmentation algorithms can be divided in the following types:

3.1 Pixel-based segmentation

Pixel-based segementation is the least complex technique to segment an image. Often a simple threshold is defined to determine whether a pixel belongs to a region or not[13]. Let I be all pixels in an image, B the resulting binary pixel value and T the threshold value:

$$B(x,y) = \begin{cases} 1 & if \, I(x,y) > T \\ 0 & otherwise \end{cases} \quad (3.5)$$

For this purpose a single threshold value is defined and for each pixel the algorithm checks if the value is within or beyond the threshold. This method is often used to create binary images. Another approach is to dismiss pixels within a certain value range from the image and keep only the pixels that are within the certain range, as illustrated in the forumla below:

$$I(x,y) \begin{cases} I(x,y) & if\, I(x,y) > T \\ 0 & otherwise \end{cases} \tag{3.6}$$

Before doing a single threshold with a single value it can be useful to smooth the image or to use more than one threshold on various region of interests. For smoothing operations, for example to reduce noise, often a 2D Gaussian filter is used, where x and y define the axis values and σ_x and σ_y the standard deviation [16]:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right) \tag{3.7}$$

2D Gaussian filter in frequency domain[16]:

$$G(u,v) = exp\left(-\frac{1}{2}\left[\sigma_x^2(2\pi u)^2 + \sigma_y^2(2\pi v)^2\right]\right) \tag{3.8}$$

One problem of pixel-based methods are if there are some various illumination effects on the image. Another problem regarding to the method of thresholding is how and where to choose the best value . If the value is to high or to low the objects may be smaller/bigger than they should be and therefore hamper the segmentation process. [13]

3.2 Contour or edge-based segmentation

If there is the necessity to avoid complex thresholding procedures or thresholding is not possible it is recommended to use edge-based respectively contour-based segmentation. An edge is defined by an extreme value of the first-order derivative or zero-crossing in the second-order derivative. Afterwards the image is being searched for local maxima in the edge strength to find the edge of an object. [13]

One downside of the algorithm is that there is no guarantee for connected edges. Thus, it is necessary to use edge linking algorithms to reconnect fragmented edges [23]. Therefore, a tracing algorithm follows the maximum edge strength around the object until it arrives at the origin again. Another drawback is that edge-based and most of the region-based (see section 3.3) methods are unlike pixel-based methods just sequential. This means that to find an edge it is necessary to process from frame to frame in contrast to pixel based segementation where all or a region of pixels can be processed parallel. Furthemore, errors in previous steps can lead to further errors in subsequent steps, which leads to the necessary of sufficient pre-processing. There are three edge-based algorithms described by Sambhunath Biswas and

Brian C. Lovell[23]:

3.2.1 Gradient operator

The gradient operator is one of the simplest edge detection algorithms. Let be e(x,y) the possible position of an edge and f the image:

$$e(x,y) = \begin{cases} 1 & if \left| \sqrt{\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}} \right| \gg 0 \\ 0 & otherwise \end{cases} \tag{3.9}$$

Thus, when the gradient operator is calculated on the location of an edge the gradient value is high otherwise if the pixel is located in the object or in the environment of the object the gradient value is low[23].

3.2.2 Laplacian operator

The Laplacian operator used the second partial derivation to calculate if the current pixel is an edge or not. The Laplacian operator indicates an edge by changing the algebraic sign. Let e(x,y) be the possible position of the edge again and f the image itself [23]:

$$e(x,y) = \begin{cases} 1 & if \left(\frac{\partial^2 f}{\partial^2 x_{i-1}} + \frac{\partial^2 f}{\partial^2 y_{i-1}} > 0 \right) \wedge \left(\frac{\partial^2 f}{\partial^2 x_i} + \frac{\partial^2 f}{\partial^2 y_i} < 0 \right) \\ 1 & if \left(\frac{\partial^2 f}{\partial^2 x_{i-1}} + \frac{\partial^2 f}{\partial^2 y_{i-1}} < 0 \right) \wedge \left(\frac{\partial^2 f}{\partial^2 x_i} + \frac{\partial^2 f}{\partial^2 y_i} > 0 \right) \\ 0 & otherwise \end{cases} \tag{3.10}$$

3.2.3 Laplacian of Gaussian Operator

D..C. Marr and E.C. Hildreth [15] proposed the Laplacian of the Gaussian operator to deal with illumination changes. Therefore, changing illumination in an image is smoothed by a Gaussian filter operation and afterwards the Laplacian operator is used to detect the edges [15].

To calculate the Laplacian of Gaussian operator the Gaussian operator 3.1 is needed, which was introduced in 3.1. Afterwards the Laplacian of the Gaussian is calculated [23], where σ_x and σ_y are the standard deviations of the axis:

$$\nabla^2 G = -\frac{1}{2\pi\sigma_x^2\sigma_y^2} \left(2 - \frac{1}{2\pi} \left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right] \right) \exp \left(-\frac{1}{2} \left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right] \right) \tag{3.11}$$

D.C. Marr and E.C. Hildreth [15] suggested a more refined procedure:

$$DOG(\sigma_x, \sigma_y) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp \left(-\frac{1}{2} \cdot \frac{x^2}{\sigma_x^2} \right) + \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp \left(-\frac{1}{2} \cdot \frac{y^2}{\sigma_y^2} \right) \tag{3.12}$$

3.3 Region-based segmentation

Region-based segmentation tries to group pixels rather on their context than only on their values. This leads to regions of group pixels with similiar characteristics. The most common initiation step of each region-based segmentation is, there is a seed point or initial region where the algorithm takes it origin. However this is also the major disadvantage of the method, because if the seed point is set to a pixel, which does not have any similiarity characterisitics with other pixels in this region, the algorithm will fail. [26]

There are two examples for region-based segmentation, which are region growing and region splitting. Region-growing needs an initial seed point and afterwards the algorithm looks for homogeneous neighbors and merges the new area to the already existing one. These steps are repeated as long as there are no more pixels to classify. The split algorithm works the other way round. First, the whole image is taken as initial seed and if the algorithm finds some inhomogeneneity, the region is splitted into subregions. This method stops if all regions are homogeneous. The region splitting aproach is less sensitive to noise than the region growing. There are also some hybrids existing like the Split, Merge and Group Algorithm to enhance the image segmentation [13].

An example for region-based segmentation, the pyramid linking algorithm, is explained at Jähne, B. [13]. The following steps are necessary for the pyramid linking algorithm:

- Calculation of the Gaussian pyramid. For instance some (gray) values of four neighboring pixels are averaged. This mean value generates a new pixel on the next level of the pyramid. This operation is similar to a box filter which smoothes the picture.

- Segmentation by pyramid-linking. This step is simple processed by assigning each pixel to either one of two pixels of the higher level, which it most likely belongs to. The decision is based on the (gray) values of the pixel and the possible two pixels of the next level. This process is done for the whole pyramid and therefore a tree model is computed, which means beginning with the pixel on the top of the pyramid, each pixel can be connected with the next lower level.

- Averaging the linked pixels. After finishing the segmentation part and building up the tree model a new mean (gray) value is computed, using the interconnected pixels. This is a bottom up operation and ends at the root of the tree.

The 2nd and 3rd step are repeated until a stable result emerges. Advantages of this method are that also very noisy pictures can be segmented successfully and there is no restriction on the form of the segmented area. A further benefit of this method is that the number of segmentation levels is not needed to be known beforehand [13].

3.4 Model-based segmentation

Model-based segmentation uses specific knowledge about the object that should be segmented like the geometric shape, their colour representation, etc. [13]. Due to the changing characteristics and object shapes of ultrasound 2D images this type of segmentation can only be used at simple use cases and therefore will not be further described.

4 Chapter 4

Change Detection Algorithms

“Change” detection or resp. motion detection tries to identify the differences of a pixel or a region between two or a set of images, see [10]. Change detection is a crucial part for the object tracking part due to the fact that without any change detection all features have to be segmented again in a picture. With motion detection it is less complex to qualify regions of interests for certain features. This chapter focuses on the main techniques according to Fatih Porikli and Alper Yilmaz in Video Analytics for Business Intelligence [10]

4.1 Frame Differencing and Background Subtraction

Frame Differencing and Background Subtraction are among of the simplest approaches. Both methods have in common that they subtract two pixels from another to obtain the differences and therefore the motion in the image. This is based on the assumption that an intensity change of a pixel inidcates a moving object. Frame Differencing can only be used for motion detection itself because if the object that moves becomes steady, the algorithm would be not able to find the object. Another major drawback is the sensitivity against illumination changes or noise. [10] Therefore it is not recommended for the modaliy of ultrasound because many coefficients influence the illumination of the image (e.g. pressure of the probe on the body, application agent). Furthermore the noise level especially of some older devices would lead to a false movement detection of random noise.

In contrast, Background Subtraction offers the possiblity of object tracking by the use of background model. Therefore the background of an image is computed to estimate the the steady regions or objects in an image (e.g. trees, streets, buildings at video surveilance images). It is important to choose a proper method to compute respectively recalculate the background model to avoid movement detection of e.g. clouds or cars which are already parked in the scene for several minutes. [10] Whereas background subtraction is a powerful motion detection and object tracking algorithm it cannot be used in the field of the ultrasound images of the kidney area. This is because there is no steady background in this area and furthermore the white noise is also a problem.

4.2 Optical Flow and Motion estimation

Optical flow is a vector field which determines the velocity and the direction of each pixel in an image. Optical flow can originate from relative motion of the observed objects or the viewer itself [11]. It describes the translation of each pixel in a region. Optical flow can be used to search for discontinutities to aid segmentation processes or to determine if some objects are moving or not [10]. Due to Horn and Schunk optical flow is computed using a brightness constraint, which assumes 'brightness constancy' of corresponding pixels in consecutive frames and is usually computed using the image derivatives. [11]. If objects respectively pixels are moving the velocity coefficient gives information how fast the observed area is moving or changing. It is important to define some image features/objects beforehand on which the optical flow is computed [10]. There are various methods existing for computing the optical flow described in [11], [7] and [8].

Despite the fact the optical flow field is very accurate and it is possible to predicte the movement of each pixel in the frame, it is not a practical method to use optical flow for time critical applications. Due to the fact that a accurate calculation of the optical flow field has high computational costs because the more accurate calculations are using iterative methods for each pixel[20]. Video compression standards like H.264 and MPEG (Moving Picutre Experts Group Standard) use some more practical methods called motion estimation techniques, which use e.g. a block-based motion estimation and compensation for the compression to predict the movement of some blocks of pixels, illustrated in Figure 4.1 [20].

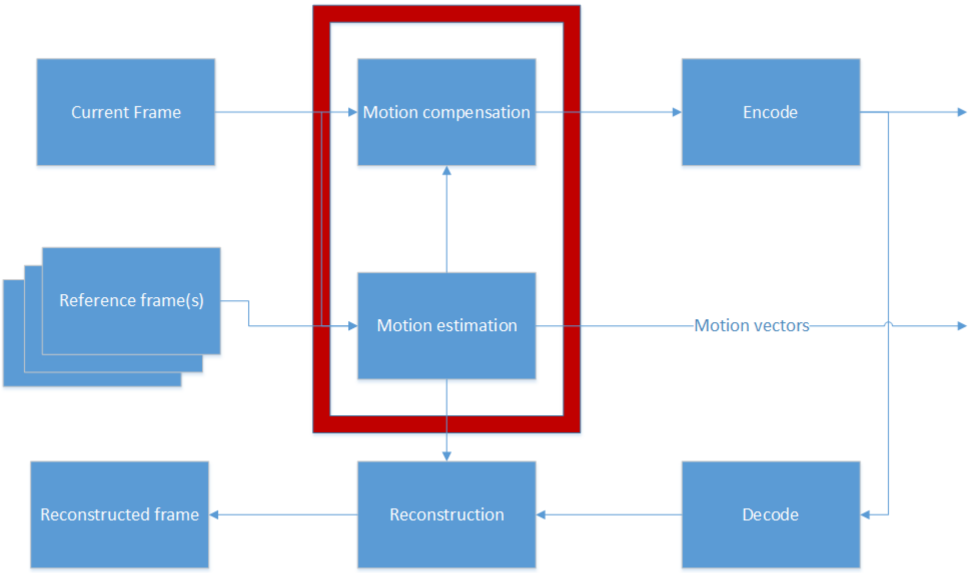

Figure 4.1: Motion estimation and compensation diagram (adapted from [21], p. 94)

5

Chapter 5

Object Tracking

This chapter deals with the methods for object tracking. It is essential that there have been some prior segmentation or motion detection methods to preprocess the image (e.g. noise cancellation, reduction), to identify the object(s) to track and to provide the tracker with prior knowledge about the object to track as the shape or pixel values. After the given object was given to the tracker it is its responsibility to find the the target in the following frames. Due to Fatih Porikli and Alper Yilmaz:

> "Robust and accurate tracking of a deforming, non-rigid and fast moving object without getting restricted to particular model assumptions presents a major challenge." ([10],page 23)

Especially the fast moving and deforming border can be a problem for most of the tracking algorithms because they assume the motion is smooth and without abrupt changes[10]. These contraints are a huge drawback for the field of the modality ultrasound and particular for the area of the kidney. Due to respiratory movements of the tissue, white noise and illumination changes the algorithms has to be either very robust to all of these issues or it is necessary to combine different approaches to assure a proper object tracking.

5.1 Object Modelling

As mentioned in the introduction of this chapter it is important to segment the object and its important features beforehand. This section deals with the transformation of the segmented features or objects to a representation form for the tracking algorithm. Due to Fatih Porikli and Alper Yilmaz described in Figure 5.1 a object model can be divided into following layers:

5.1.1 Model Representations

The representation depends to the field of use. According to the chosen model it either limits or extends the possible types of motion or deformation[10]. For the field of the kidney area in US-images the most appropriate representations would be the object region (e.g. ellipse or square) or the silhouette.

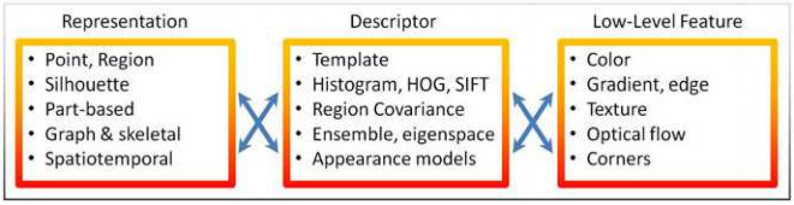

Figure 5.1: Layers of the object model (taken from [10], p. 13)

5.1.1.1 Region

A region can be represented by a shape around its centroid or a set of points. Motion detection for such shapes can be done by simple translation, similarity and homography or affine transformations. This model can be used for rigid or non-rigid objects [10].

5.1.1.2 Silhouette

The silhouette is a region inside the border of a object. The most frequently used silhouette form is a simple binary approach, where ones stands for the object region and zeros for regions outside the object. Furthermore there is a explicitly or implicitly representation form for edge-based methods. Explicit is defined by a set of support points. These points are interconnected via spline equations[10]. An implicitly representation would be a level set [25]. The level set has the ability to adpat to topology changes and is extendible to higher dimensions without any changes in its formulation [10].

5.1.2 Model Descriptors

Descriptors allow to describe regions of interest (ROI) in a mathematical way. Therefore all coefficient of the object or the image are important for the discriptor like the size, range, noise and image artifacts. Descriptors are more distinct the bigger the ROI is. Most descriptors have a lack of a proficient similiarity criterion which registers statistical and spatial properties[10].

5.1.2.1 Template

Templates are one of the most common used descriptors and use the model representation of regions or silhouettes. One major drawback of templates is the lack of reacting to changes of the viewers perspective or object changes. Therefore templates are hard to use for the modality of ultrasound, as the 2D representation of a object can transform in a very short time if the focus of the probe moves to another layer of the tissue. Furthermore due to normal anatomic variations and various imaging modes the kidney appears in different shapes, colours and sizes.

5.1.2.2 Histogram

A histogram estimates the probability distribution of certain features within a given ROI. There are serveral factors that can be considered for the computation of the histogram like raw colour values, derivative information or texture measures. To compute a histogram first the quantization levels are defined and afterwards the number of pixels which belong to the certain quantization levels are counted. Often there has to be done a mean colour adjustment prior to remove illumination and shadowing effects [10].

There are some related approaches named Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG), which use image gradient instead of colour values. SIFT is invariant to rotation and is more resilient to image deformations [10].

Another possiblity for histograms is the shape histrogram. For this purpose the spatial relation between a reference point on the contour of the shape to other points is modeled by a histogram. The relation is measured by computing the angle and magnitutde of the vector which joins both points. By taking random or all points of the contour a set of histograms is generated, which forms a distribution based descriptor [10].

5.1.2.3 Covariance

The covariance matrix allows to fuse multiple features. The diagonal entries represent the variance of the features and the remaining entries the correlations among them.

5.1.3 Model Features

Model features are an important coefficient for the tracking performance, depending on the set of features used for tracking purposes. In general the features which distinguishes best between background and object or between multiple objects are also the best for tracking purposes. The main challenge is, to choose the correct features and the combination and their weight online to aid the tracking algorithm best. In most tracking algorithms the features are chosen manually by the user. Nevertheless, automatic feature selection is also possible. Those can be divided into filter methods, which try to select the features based on a general criteria, whereas wrapper methods select the features based on a specific problem domain. Examples for filter methods are the Principial Component Analysis (PCA) and for wrapper methods boosting. In boosting the algorithm discovers a weightet combination of feature classifiers to maximize the classification performance[10]. Due to [10] there are the following visual features:

- Colour: is influenced by the the spectral power distribution of the illuminant and the surface reflectance properties of the object. The most common used color space in imaging is the red, green, blue (RGB) space. One problem of the RGB space is, that the colour differences to which humans are sensitive are not corresponend to the differences between the colours in RGB. Other colour spaces like YUVand LAB do not have this issue. Instead the HSV colour space is only approximately uniform. All these colour spaces are sensitive to noise and therefore it cannot be estimated which colour space is more efficient or the best, which is noticeable due to the fact that a variety of colour spaces are used for tracking purposes[10].

- Gradient: For determining the gradient often edge-based segmentation algorithms are used as introduced in Chapter 3.2. An important advantage of gradients is the less sensitivity against illumination changes in contrast to to colour features [10].
- Optical Flow: As introduced in the “Movement Detection Algorithms” 4.2 optical flow is computed using a brightness contraint and uses the image derivatives. Optical flow is used as a feature in motion-based segmentation[10].
- Texture: Measures the intensity variation of a surface. In contrast to colour features, textures need a descriptor generated via preprocessing. Some texture discriptors are gray level co-occurence matrices, wavelets, Gabor filters or steerable pyramids. As gradient features, texture is more insensitive to illumination changes [10].
- Corner Points: Corner points have a low computational complexity and are simple to implement, therefore corner points are one of the most commonly used features. As an exmpale the Harris corner detector works by guessing whether the colour value of pixels in a certain area of a point of interest is high or not[10]:

$$E(x,y) = \sum_{u}\sum_{v} \left(I(x+u, y+v) - I(x,y)\right)^2 \tag{5.1}$$

Taylor series approximation around (x,y), where the matrix (M) defines an ellipse with minor and major axes denoted by its eigenvalues and the extent by the eigenvalues:

$$E(u,v) = [u\, v] \underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{M} \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.2}$$

This texture content is computed for all pixels and weak interest points or multiple candidates in a small neighborhood are removed by nonmaximal supression. The Harris detector provides feature points at multiple scales which can be combined to make these points scale invariant. Drawbacks of the Harris detector are the lack of location possiblities of interest points at the level of subpixels and setting the number of interest points it detects[10].

5.2 Object Tracking

Once one or more target objects have been segmented and detected in an image a tracker's algorithm tasks is, to identify the object(s) in the consecutively frames. Some important questions for object algorithms are:

- Which features should be used to track the object(s)?
- What object model is suitable for this case?

It mostly depends on the image modality and the use case which method of object tracking is the most appropriate strategy. Following from a recital of Fatih Porikli and Alper Yilmaz [10] there are the following object tracking algorithms available

5.2.1 Template Matching

A simple approach to detect a subimage of an image in an consecutively image [1].

Template matching is processed by searching a region or shape in an image which is similiar to a region or shape in a previous one. This method is one of the most common used methods for object tracking and the template is defined by some simliarity measures where intensity features, color features or image gradients. The feature selection depends on whether the feature has to be resistent to illumination changes or not(compare Section 5.1.3) [10]. The template is set to the new image and the algorithm checks how many points of the image match with the template. This approach is repeated throughout the image and every pixel. In the end the point with the best machting result is presented as the result [1]. Due to this "brute force" search computational costs are very high and can be reduced by defining a search window, where to search for the corresponding object in the next frame [10]. Another drawback of the basic template matching algorithm is its lack to deal with rotation and scale changes. Indeed the algorithm is position invariant however, for scale and rotation invariance the algorithm has to be enhanced to the Mellin transform, for scale invariance, or the Fourier-Mellin transform to tackle both tasks [1].

5.2.2 Density Estimation

There are some various methods for density estimation. One of the most common approaches is the mean-shift. There are some variations for the mean-shift which will be explained in this section:

5.2.2.1 Mean-Shift

Due to Fatih Porikli and Alper Yilmaz [10] the Mean-Shift algorithms is defined as:

> " Mean-shift is a nonparametric density gradient estimator to find the image window that is most simiar to the object's color[sic] histogram in the current

frame." ([10], page 24)

The basic algorithm due to [6], with a set of n image points $\{x_i\}_{i=1\ldots n}$in the space $R^d$with d dimensions, window radius h and the multivariate kernel density estimate with the kernel K(x) [6]:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{5.3}$$

The multivariate Epanechnikow kernel [6] is calculated, where $c_d$is the volume of the unit d-dimensional sphere.:

$$K_E(x) = \begin{cases} \frac{(d+2)(1-\|x\|^2)}{2c_d} & if\ \|x\| < 1 \\ 0 & otherwise \end{cases} \tag{5.4}$$

By introducing a function k from kernel K, where K(x) = $k(||x||^2)$ the density changes to following:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \tag{5.5}$$

and formalise a new profile:

$$g(x) = -k'(x) \tag{5.6}$$

and presuming that for all $x\epsilon[0, \infty)$, except for a finite collection of image points, a derivative of k is existing. The kernel G can be defined, where C is a normalisation constant:

$$G(x) = C_g(\|x\|)$$

The mean shift vector can after the evaluation of the density gradient be written as follows:

$$M_{h,G}(x) \equiv \frac{\sum_{i=1}^{n} x_i g\left(\|\frac{x-x_i}{h}\|^2\right)}{\sum_{i=1}^{n} g\left(\|\frac{x-x_i}{h}\|^2\right)} - x \tag{5.7}$$

The mean shift algorithm is based on a recursive calculation of the mean shift vector $M_{h,G}(x)$ and adjusting the centroid of the kernel G by $M_{h,G}(x)$[6].

Based on Gary R. Bradski[5] the mean shift is executed as follows:

- Define a search window size.
- Set the initial location of the search window.
- Mean Location is computed in the search window.
- Mean location defines the new center of the search window.

- Iterate through Step 3 and 4 until there are no or only minor changes in the mean location (convergence).

Mean shift tries to find a target model within a image with the aid of the simiarity which is expressed by an metric based on the Bhattacharyya coefficient [6]. This is done by using a histogram of the object to track ($\mathrm{h_o}$) and comparing it to the histogram of the window around the possible object location ($\mathrm{h_c}$). Bhattacharya distance is used to define the histogram distance. So either minimizing the Bhattacharya distance or maximizing the Bhattacharya coefficient [10]

$$p(h_o, h_c) = \sum_{b=1}^{B} [h_o(b) h_c(b)]^{\frac{1}{2}}$$

and afterwards building the Taylor series and compute the likelihood ratio between the object model histogram ($\mathrm{h_o}$) and the candidate histogram ($\mathrm{h_c}$). The candidate window will be shifted towards the direction, where the Bhattacharya coefficient is a maximum or the Bhattacharya distance is a minimum in every iteration step. The computation process stops if there are no significant changes at the Bhattacharya distance or coefficient [10].

In short at every computation step the algorithm tries to raise the histogram similarity and stops if there are only minor changes left so the value is pending around a limit. Usually it takes around five to six iterations until the computation is finished. By using a weighting scheme for the histogram calculation, which considers pixels that are located nearer to the object center more, enhances the mean shift tracking algorithm.

One of the major advantages of the mean shift approach contrary to the template matching are the lower computational costs and the prevention of a brute-force search. But the algorithm will fail if the search window does not covers the object to track in the new image at all [10]. Following there are some variations of the mean shift for performance, robustness enhancement for certain use cases.

5.2.2.2 CAMSHIFT

“Continuously Adaptive Mean Shift” (CAMSHIFT) and was first introduced by Gary R. Bradski [5] which is an adaption of the mean shift algorithm 5.2.2.1. CAMSHIFT is modified to deal with dynamically changing colour probability distribution in video frame sequences. This issue arises if some object in a video sequence is being tracked and moves so that size and location of the probability distrubtion changes. CAMSHIFT always adjusts its search window size by using the zeroth moment information which is a part of the algorithm [5]. CAMSHIFT was introduced as a algorithm to detect faces or other parts of flesh coloured parts of the human body in video sequences. Therefore, the HSV colour model was proposed instead of the RGB colour model. This is because all human races, except for albinos, have the same hue level and therefore leads to better results in terms of tracking human flesh in video sequences. Due to Gary R. Bradski [5] CAMSHIFT is calculated as follows:

- Set the initial location of the search window.

Algorithm 5.1 Pseudocode CAMShift

```
SearchWindow = [InitialPosition , InitialSize]
convergence = false;
meanLocation[];
int i = 0;
while(convergence)
{
        [meanLocation(i),zerothMoment] = meanShiftAlgorithm();
        SearchWindow[Position] = RecalculatePosition(meanLocation);
        SearchWindow[Size] = RecalulateSize(ZerothMoment)
        if(DetermineChanges(meanLocation) == <<)
        {
                convergence = true;
        }

}
```

- Use mean shift iterations introduced in Section 5.2.2.1 + store zeroth moment.
- Calculate the new search window size by a function of the zeroth moment
- Iterate through Step 2 and 3 until there are no or only minor changes in the mean location (convergence).

The mean location or centroid can be calculated through the following formula, proposed by Gary R. Bradski [5], where I(x,y) is defined as the pixel value at position of x and y in the image. x and y defines the range over the search window:

Calculate zero$^{\text{th}}$ moment:

$$M_{00} = \sum_x \sum_y I(x, y) \tag{5.8}$$

Find first moment (x and y):

$$M_{10} = \sum_x \sum_y xI(x, y) \tag{5.9}$$

$$M_{01} = \sum_x \sum_y yI(x, y) \tag{5.10}$$

Calculate centroid:

$$x_c = \frac{M_{10}}{M_{00}} \tag{5.11}$$

$$y_c = \frac{M_{01}}{M_{00}} \tag{5.12}$$

Some advantages of CAMSHIFT against mean shift are outlined in the paper of Gary R. Bradski [5]:

- CAMSHIFT can handle irregular object motion due to scaling of the search window to the object size.
- CAMSHIFT can also handle noise, because of the usage of a colour model and outliners are mostly ignored by the algorithm.
- Some objects with the same colour as the object to track cannot distract the algorithm if they are outside the search window.
- If the object does not disappears to 100% the CAMSHIFT algorithm will track the remaining parts of the object.
- Due to the usage of the HSV colour model, CAMSHIFT is more tolerant against illumination changes than the mean shift algorithm.

CAMSHIFT is a good tracker if e.g. faces should be tracked in a video sequence. Due to the usage of the colour tracker and the HSV colour model it is even very stable against noise. One major drawback of the algorithm is that it is built on colour distributions alone. So if there is any error in the colour model, the tracker will fail to identify the object properly [5].

This drawback is critcal to the modality of ultrasound as in ultrasound original images differences beetween objects often only can be estimated by the different brightness level. A possiblity to ease this issue would be a further preprocessing step, where different regions are coloured in various colours. Only regions which are interconnected possess the same colour. This method requires a very proper preprocessing to assure that false interconnections do not occur, otherwise the algorithm will fail again.

5.2.2.3 Scale and orientation adaptive mean shift tracking

Due to the problem that the basic mean shift tracker is not able to handle changes of scale and orientation of the target object the Scale and orientation adaptive mean shift tracking (SOAMST) algorithm was proposed by J. Ning, L. Zhang, D. Zhang and C. Wu [12]. SOAMST works as follows [12]:

- Calculation of the target model $\hat{q}$ and intialisation of the position y_0 of the model in the preceding frame.
- Initialisation of the iteration number $k \leftarrow 0$.
- Calculation of the target candidate model $\hat{p}(y_0)$ in the current frame.
- Calculation of the weight vector $\{w_i\}_{i=1\cdots n}$.
- Calculation of the target models new position y_1 with the algorithm in Section5.2.2.1

- Let $d \leftarrow \|y_1 - y_0\|$, $y_0 \leftarrow y_1$ and set the error threshold ε and the maximum number of iterations N.

  If(d<ε or k $\geq$ N) => Stop and go to the next step

  Otherwise => $k \leftarrow k+1$ and start over at the calculation of the target candidate model in the current frame

- Get the width, height, and orientation from the target candidate model with the aid of the covariance matrix.

- Estimate the initial target model for the next frame.

Whereas scale and orientation issues are already covered by the CAMSHIFT algorithm (compare Section 5.2.2.2). However, due to the use of the HSV model in the basic algorithm CAMSHIFT is not the first choice for object tracking in ultrasonic images.

Although the SOAMST algorithm can handle scale and orientation changes, it will fail if the changes are abruptly in consecutively frames of the video. One approach of SOAMST is to enlarge the window size for the target candidate region. Therefore, the target object is inside the search window no matter if its scale or orientation has changed smoothly[12].

5.2.3 Regression

Regression assumes that there is some relationship between variables which can be modeled. So for describing the approach mathematically [10] defined some pairs (α_i, $\mathrm{X_i}$) of data $\alpha\epsilon\mathbb{R}^d$ in vector space. Furthermore, there is a need of some points on the manifold $X\epsilon M$. So the regression algorithm φ maps the vector space data onto the manifold[10]:

$$\varphi : \mathbb{R}^d \longmapsto M$$

First the sum of the squared geodesic distances between the point $\mathrm{X_i}$ and the estimations $\varphi(\alpha_i)$ is calculated:

$$Reg = \sum_i \triangle^2 \left[\varphi(\alpha_i),\, X_i\right] \tag{5.13}$$

By using Lie algebra on the manifold, the algorithm can be approximated as

$$Reg = \sum_i \left\|\log\left[\varphi^{-1}(\alpha_i)X_i\right]\right\|^2 \approx \sum_i \left\|\log\left[\varphi(\alpha_i)\right] - \log\left[X_i\right]\right\|^2 \tag{5.14}$$

till to the first order terms. The regression function φ can also be expressed as, where $\Omega : \mathbb{R}^d \mapsto \mathbb{R}^r$, which is the d x r matrix of regression coefficients, assumes the tangent vectors $\log(\mathrm{X_i})$ on the Lie algebra:

$$\varphi(\alpha_i) = \exp(\alpha_i^{\mathrm{T}}\Omega) \tag{5.15}$$

So the function becomes:

$$Reg = \sum_i \left\| (\alpha_i^{\mathrm{T}} \Omega) - \log [X_i] \right\|^2 \tag{5.16}$$

Now we define a k x d matrix of initial observations X and the k x r matrix of corresponding mappings to the Lie algebra Y:

$$X = \begin{bmatrix} [\alpha_1]^{\mathrm{T}} \\ \vdots \\ [\alpha_k]^{\mathrm{T}} \end{bmatrix} \quad Y = \begin{bmatrix} [\log (X_1)]^{\mathrm{T}} \\ \vdots \\ [\log (X_k)]^{\mathrm{T}} \end{bmatrix} \tag{5.17}$$

Those two matrices filled into the Reg function replaces the summation with a trace:

$$Reg = tr \left[(X\Omega - Y)^{\mathrm{T}} - (X\Omega - Y) \right] \tag{5.18}$$

When the Reg function is differentiated in respect to Ω, a minimum is at $\left(X^{\mathrm{T}} X\right)^{-1} X^{\mathrm{T}} Y$. Further improvements can be done to avoid overfitting [10].

Once the regression algorithm starts, the regression function for the object is estimated. These function maps the region feature vectors to the assumed affine motion vector. At the tracking procedure the feature vectors are extracted from the previous object location and applied for the training of the regression function[10].

5.2.4 Motion estimation

Motion estimation uses optical flow methods, which were introduced in Section 4.2. To use optical flow as a tracker it is necessary, that the tracker is able to handle translations e.g. of a rectangluar region[10]. Therefore the Kanade-Lucas-Tomasi (KLT) tracker can be used to compute the translation (du, dv) of a region, which is centered on an interest point. The first part of the formula was already introduced in Section 5.1.3:

$$\begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix} \tag{5.19}$$

After the recalculation of the interest point the KLT tracker starts to evaluate the quality of the tracked region, by calculation the sum of square differences between the projected region and the current region. If the result of the calculation is small the feature is used for further tracking, otherwise it will be eleminated[10].

5.2.5 Kalman Filtering

If $\mathrm{X_t}$ defines the location state sequences of a moving object and the linear system influences the state in time, where η is defined as white noise with covariance $\sum^{\eta}$[10]:

$$X_t = A_t X_{t-1} + \eta \tag{5.20}$$

The measurement equation describes the connection between the measurement and the state, where D_t is the measurement matrix and ξ is the white noise with covariance $\sum^{\xi}$ independent of η:

$$Y_t = D_t X_t + \xi \tag{5.21}$$

For tracking purposes either the probability density function or the estimation of the state X_t with all the measurements up to the current moment is used. The Kalman filter estimates the state of a linear system, but the distribution of the state has to be Gaussian. To calculate the new state, a state model is used by Kalman filtering [10], where Σ_t^p is the covariance predictions at time t and $X_t^p$are the state predictions at time t. A_t defines a matrix which includes the connection between the state values at time t and $t-1$:

$$\begin{aligned} X_t^p &= AX_{t-1} && (5.22)\\ \Sigma_t^p &= A\Sigma_{t-1}A^{\mathrm{T}} + \Sigma_t^{\eta} && (5.23) \end{aligned}$$

Afterwards the object's state is updated by using the current observations Y_t, where G_t is the Kalman gain to propagate the models of the state[10]:

$$\begin{aligned} X_t &= X_t^p + G_t\left[Y_t - D_t X_t^p\right] && (5.24)\\ G_t &= \Sigma_t^p D_t^{\mathrm{T}}\left[D_t \Sigma_t^p\right] && (5.25)\\ \Sigma_t &= \Sigma_t^p - G_t D_t \Sigma_t^p && (5.26) \end{aligned}$$

Before Kalman filtering can be used on multiple objects it is necessary to correspond the most likely measurement of a object to that object's state, which can be achieved by using the nearest neighbor approach. But the algorithm still fails to converge, if the objects are to close to each other and a missmeasurement has been performed.

5.2.6 Particle Filtering

To overcome the limitation, that only a Gaussian distribution can be handled by the Kalman filter, particle filtering has been introduced. The disadvantages of particle filerting is the sample degeneracy and impoverishment, if the importance sampling is used, which is introduced in this subsection. The drawbacks are particularly noticeable, if the representation is high dimensional. The conditional state density $p(X_t|Y_t)$ in particle filtering at time t is using a set of $N_s$samples, wich are the particles $\{s_{t,n}\}$where n=1, .., N_s with weights $\pi_{t,n}$. The weights are influenced by the sampling probability and define the weight or importance of a sample. A cumulative weight c_n, whereas $\sum c_n = 1$ is also used. Every new sample at time t derive from $S_{t-1} = \{(s_{t-1}, \pi_{t-1}, c_{t-1}, n)\}$from t-1[10]. The most coomond sampling scheme due to [10] is the importance sampling with the following steps:

- Selection of random samples
- For each selected sample a new sample will be predicted.
- Afterwards the weights are updated corresponding to the new samples.

With the aid of the new samples the object position can be estimated. Furthermore, resampling is used to eleminate samples with very low weights, to retrieve always the best particles. To track multiple objects [10] mentioned the Joint Probability Data Association Filter and the Multiple Hypothesis Tracking, whereas the Joint Probability Data Association Filter is not suitable for the use case of kidney stone tracking due to the inability to handle objects which leave the image or new objects entering in the following frames [10].

5.2.7 Silhouette Tracking

Silhouette tracking operates with the aid of object models which are generated from previous frames. There are various types of silhouette tracking in differences in:

- features used to for tracking
- occlusion handling
- if there is a training needed or not.

Advantages of silhouette trackers are there ability to deal with various types of shapes, their robustness against noise. Fatih Porikli and Alper Yilmaz [10] focussed on two types of silhouette tracking:

5.2.7.1 Shape Matching

In shape matching first the object silhouettes are extracted and afterwards in the consecutively frames the object is tracked by computing distance values between the model and the object to track. One drawback of this method in terms of the modality ultrasound and the specific use case of this thesis is that nonrigid object motion is usually carried out by background subtraction. As already mentioned in 4.1 this method nearly useless due to the lack of a existing background.

5.2.7.2 Contour Evolution

For contour evolution it is vital for the algorithm that a part of the object in the current frame overlaps with the region tracked or segmented in the previous one. There are two approaches for the contour evolution algorithm. The first one uses state space models for the contour shape and motion. The second uses direct minimization techniques like gradient descent to minimize the contour energy and therefore evolving the contour.

5.2.8 Active Contours

Refering to A.Blake in [17] an active contour is:

> "... a parameterised curve r(s), $0 \leq s \leq 1$in the plane that is set up to the attracted features in an image I(r)." ([17], page 296)

Andrew Blake and Michael Isard [2] mentioned that an active contour use prior information about objects to segment and track them. Moreover, active contours do not process the whole image but rather to pre selected regions of the image, which enhances the speed and efficiency of the algorithm and enables the possibility to process videos at a full rate of 50/60 Hertz (Hz).

Appyling active contours respectively active shape models can vary from snakes, deformable templates and daynamic contours. To use active contours for tracking it is necessary to deal with inertia, restoring forces or damping[2]. In the following points listed there are types of active contours described and discussed, which can be used for object tracking purposes:

5.2.8.1 Snakes

The invention of snakes was based on a problem of feature detection. The principle of feature detection is a mask, which has a high pixel value for that position, where a strong prescence of a chosen feature occured. Afterwards there is a new image computed, which is also called feature map. These feature map only enhances the chosen types of features but does not detect them. For detection often a threshold was used, but a constant value threshold causes many problems like gaps in edges if the threshold is to high or spurious edges if the threshold is to low. Even other low-level feature processes where explored but they were not very accurate in retrieving entire geometric structures[2].

Therefore, the invention of snakes was kind of revolutionary, because snakes do not use the by then common bottom up approach to detect features or objects. Instead they used the top down approach. The active contour method of snakes work with a potential energy field F(r), which is a function of the image intensity landscape. The snake r(s), which is a deformable curve, where $0 \leq s \leq 1$ (as introduced in the citation of [17] at Section 5.2.8) is positioned in this field and if $F(r) = -|\nabla I|$ the snake would be attracted against high image contrast. The equilibrium equation of the snake is defined as following, where ∇F is the spatial gradient of F and denotes the potential energy of the dynamical system, ω_1 and ω_2 are responsible for the restoring forces, which are the elasticity and stiffness[17]. The external force is counterbalanced by the internal potential energy, which is responsible to define the smoothness of the shapes respectively curves[2]:

$$\underbrace{\left(\frac{\partial\left(\omega_1 r\right)}{\partial s}-\frac{\partial^2\left(\omega_2 r\right)}{\partial s^2}\right)}_{internal\, forces}+\underbrace{\nabla F}_{external\, force}\quad=\quad 0 \tag{5.27}$$

Changes to ω_1 and ω_2 affects the smoothness and shape of the curve, e.g. if ω_2 spirals downward 0 the snake will kink at the point $s = s_0$. Increasing ω_2 would make the snake more smooth, but also increases the chance to regress the snake towards a straight line. Whereas increasing $\omega_1$makes the snake more like a strechted elastic, which ease an even parameterisation of the curve but also raises the chance that the snake lines are getting to short or are even collapsing to a small point if they are unbalanced to the external energy or contraints[2]. Afterwards to compute this system in finitely time for practical use cases it is necessary to approximate the continuous trajectories in the formula[17]. Andrew Blake and Michael Isard [2] suggested the usage of finite differences to approximate the spatial derivatives r_s and r_{ss}, where h defines some intervals which seperate the original sequence of samples of r(s) :

$$r_s(s_i) = \frac{r(s_i) - r(s_{i-1})}{h} \tag{5.28}$$

$$r_{ss}(s_i) = \frac{r(s_{i+1}) - 2r(s_i) + r(s_{i-1})}{h^2} \tag{5.29}$$

Afterwards the system can be solved in time $\mathcal{O}(N)$[2].

5.2.8.2 Dynamic Contours

To use the Snakes for shape tracking it is necessary to adjust its behaviour over an image sequence rather than a single image I(r). Therefore, I(r,t) must be defined. This leads to a Langrangian dynamical system [17], which adds γ for the viscosity of the medium and ρ for the distributed mass along the contour:

$$\underbrace{\rho r_{tt}}_{inertial\, force} = \quad -\underbrace{\left(\gamma r_t - \frac{\partial\left(\omega_1 r\right)}{\partial s} - \frac{\partial^2\left(\omega_2 r\right)}{\partial s^2}\right)}_{internal\, forces} + \underbrace{\nabla F}_{external\, force} \tag{5.30}$$

This formula represents Newton's law of motion and represents the basic of the dynamic contour. Now the dynamic contour becomes some kind of two-phase process, which fuses prediction with measurements. The two-phase process contains a dynamical model, which predicts the motion from one frame to another. Afterwards this estimated positions for the next frame is refined by measured image features. Kalman filtering, as introduced in 5.2.5, can be used for this two-phase process [2]. For practical implementation, all the parameters ω_1,ω_2,γ and ρ have to been chosen wisely and may have to be spatial functions instead of constants only. Therefore probabilisitc temporal filtering, like Kalman filtering, can be used, which demands much lower dimensional state space to reduce computational efforts and gain stability [17]. There is a need of probabilistic treatment for predictive models, because otherwise the prediction would be to strong. Furthermore, if the predictions were only deterministic without any room for uncertainty the measurements would be ignored, because the prediction would overrule the measurements[2].

6

Chapter 6

Conclusion

In this report several algorithms for object tracking were presented and discussed in Chapter 5. Due to some theoretical facts about the algorithms and some basic implementations there are two recommendations for the image modality of ultrasound:

- Mean-Shift and Mean-Shift derivations: There are various implementations of the Mean-Shift algorithm and some were discussed in this report like the small derivation of the basic Mean-Shift in Section 5.2.2.1, the CAMShift in Section 5.2.2.2 and the SOAMST in Section 5.2.2.3. The examination of this algorithms showed that the Mean-Shift and the SOAMST algorithm are suitable to address object tracking issues in ultrasonic images. Both algorithms operate in a stable manner, aided by sufficient pre-processing steps. However, the CAMShift algorithm would need a different kind of pre-processing to use it in other use cases than tracking human flesh like changing of the colour model.

- Active Contours: Active contours were examined in Section 5.2.8. Some small tests showed that Active Contours are easy to implement and fast in processing. The examination showed also that the algorithm is able to detect objects although the size of the starting model of the object to track is smaller or bigger than the actual object. Furthermore, the algorithm is also able to handle object movement it is only important to choose appropriate settings for the algorithm, depending on the image modality, application, quality of the images and the methods of pre-processing used.

Both alogrithms have in common that there are several frame works and implementations available for various programming languages. Implementations and frame works of the algorithms are inbuilt features of development environments too (e.g. MATLAB[1]). It is possible that there are some further algorithms for the modality of the ultrasound which are maybe equal or better than the Mean-Shift and the Active Contour algorithms, however the work on this report showed that these two principles are robust, easy to implement and less complex compared to the other methods discussed in Chapter 5.

[1] http://www.mathworks.com/products/matlab/

Bibliography

[1] Mark Nixon; Alberto S Aguado. *Feature Extraction & Image Processing.* Academic Press, 2nd edition, 2008.

[2] Michael Isard Andrew Blake. *Active Contours The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion.* Springer, 1998.

[3] Alan C. Bovik. *Handbook of Image and Video Processing.* Academic Press, 2000.

[4] Alan C. Bovik. *The Essential Guide to Image Processing.* Academic Press, 2009.

[5] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal No. Q2, 1998.

[6] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. Technical report, Electrical & Computer Engineering Department, Rutgers University, 2000.

[7] Y. Weiss D. Fleet. *Handbook of Mathematical Models in Computer Vision.* Springer US, 2006.

[8] Prof. Daniel Cremers Dr. Andreas Wedel. *Stereo Scene Flow for 3D Motion Analysis.* Springer London, 2011.

[9] Wim C. Van Etten. *Introduction to Random Signals and Noise.* John Wiley & Sons, 2005.

[10] Alper Yilmaz Fatih Porikli. *Video Analytics for Business Intelligence*, volume 409, chapter Object Detection and Tracking, pages 3–41. Springer Berlin Heidelberg, 2012.

[11] Schunck Brian G Horn, Berthold K.P. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[12] D. Zhang J. Ning, L. Zhang and C. Wu. Scale and orientation adaptive mean shift tracking. *IET Computer Vision*, 6:52–61, 2012.

[13] Bernd Jähne. *Digital Image Processing.* Springer Berlin Heidelberg, 6th edition, 2005.

[14] Marco A. Serpa Lopez. Suitability of tumour tracking for the verification of respiratory gated radiation therapy. Master's thesis, University of Canterbury, 2011.

[15] D.C. Marr and E.C. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, February 1980.

[16] Mohamed Najim. *Digital Filters Design for Signal and Image Processing.* John Wiley & Sons, 2006.

[17] Yunmei Chen Nikos Paragios and Olivier Faugeras, editors. *Handbook of Mathematical Models in Computer Vision.* Springer US, 2006.

[18] Roger Bourne PhD. *Fundamentals of Digital Imaging in Medicine.* Springer London, 2010.

[19] George P. Rédei. *Encyclopedia of Genetics, Genomics, Proteomics and Informatics.* Springer Netherlands, 2008.

[20] Iain E. Richardson. *The H.264 advanced video compression standard.* John Wiley & Sons, 2010.

[21] Iain E. G. Richardson. *Video Codec Design: Developing Image and Video Compression Systems.* John Wiley & Sons, 2002.

[22] Giorgio Rizzatto. Ultrasound transducers. *European Journal of Radiology*, 27:188–195, 1998.

[23] Brian C. Lovell Sambhunath Biswas. *Bézier and Splines in Image Processing and Machine Vision.* Springer London, 2008.

[24] Dietrich Schlichthärle. *Digital Filters Basics and Design.* Springer Berlin Heidelberg, 2nd edition, 2011.

[25] J. Sethian. Level set methods: evolving interfaces in geometry, fluid mechanics, computer vision and material sciences. *Cambridge University Press*, 1999.

[26] Lilly Spirkovska. A summary of image segmentation techniques. Technical report, Ames Research Center, Moffett Field, California National Aeronautics and Space Administration (NASA), June 1993.