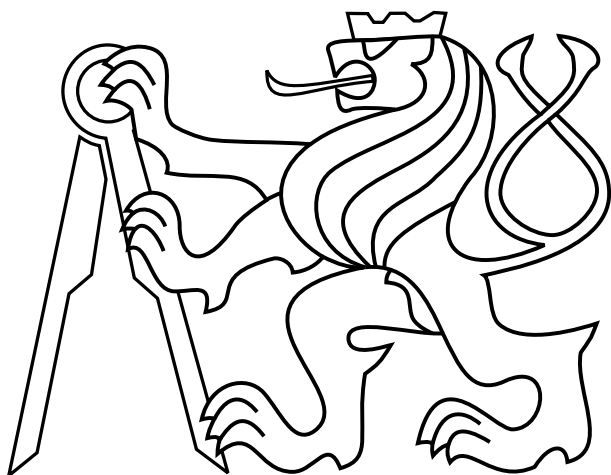


**CZECH TECHNICAL UNIVERSITY
IN PRAGUE**



DOCTORAL THESIS STATEMENT

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Ing. Jan Dupač

Stable wave detector and tracker

PhD Study Programme No. P 2612—Electrotechnics and Informatics,
branch No. 3902V035—Artificial Intelligence and Biocybernetics

Doctoral thesis statement for obtaining the academic title of “Doctor”,
abbreviated to “PhD”.

Prague, 25. února 2011

The doctoral thesis was produced in full-time PhD study at the department of Cybernetics of the Faculty of Electrical Engineering of the CTU in Prague.

Candidate: Ing. Jan Dupač
Department of Cybernetics,
Faculty of Electrical Engineering,
Czech Technical University in Prague

Thesis Advisor: Prof. Ing. Václav Hlaváč, CSc.
Department of Cybernetics,
Faculty of Electrical Engineering,
Czech Technical University in Prague

Opponents:

The doctoral thesis statement was distributed on:

The defence of the doctoral thesis will be held on at a.m./p.m. before the Board for the Defence of the Doctoral Thesis in the branch of study No. 3902V035 in the meeting room No. of the Faculty of Electrical Engineering of the CTU in Prague.

Those interested may get acquainted with the doctoral thesis concerned at the Dean Office of the Faculty of Electrical Engineering of the CTU in Prague, at the Department for Science and Research, Technická 2, Praha 6.

Prof. Ing. Vladimír Mařík, DrSc.
Chairman of the Board for the Defence of the Doctoral Thesis
in the branch of study No. 3902V035—
—Artificial Intelligence and Biocybernetics,
Department of Cybernetics, Karlovo náměstí 13, Prague 2

Abstract

A new salient semantics-less blob detector named the Stable Wave Detector (SWD) is suggested. The detector proved to be particularly very useful in tracking a target in a videosequence. SWD is a multiscale blob detector in the intensity image. The targets are blobs in 2D which correspond to local maxima/minima of the image intensities or positive and negative peaks in 1D. SWD belongs to a group of interest point/regions-like operators aiming at detecting repeatedly distinguished entities regardless of their meaning. Precision, robustness and speed of the SWD detector are main issues. The detector is based on the phase of the first harmonic wave in the window sliding across the image. Its localization is a result of an integral transformation rather than a usual derivative. Thus, the blob detector is inherently robust to noise and blur in the image. The SWD provides a subpixel localization of blobs together with the estimate of its precision, the measure of the strength/significance and the estimate of the size/scale for each blob. Detected blobs have some minimal distance given by the period used for detection. Their minimal mutual distance can facilitate matching of corresponding scene points. The properties of the SWD detector itself did not beat the state of the art considerably.

The situation has improved dramatically when the SWD detector was employed in tracking a target, i.e. in establishing a point to point (blob to blob) correspondences across the videosequence. The outcome is an ultrafast tracker able perform better than the state-of-the-art. The tracker takes two inputs: (a) the location of the blob in the previous frame estimating its location in the current frame and (2) the current frame in which the blob should be found. The problem of tracking is considered in several levels of abstraction. At a low level, correspondences between individual points are established at different scales independently. The high level of tracking system incorporates a model of the movement and a model of the target. The trade off between the precision and the range of trackable displacement at different scales must be solved. It is shown how the movement and target model can be used in a coarse to fine approach solving the trade off between the precision and the range.

The SWD detector and the tracker utilizing it were extensively tested experimentally. Test results are presented in the thesis. The implementation of the involved algorithms in C language is provided to the research community.

Obsah

1	Motivation and problem formulation	1
1.1	Motivation	1
1.2	Problem formulation	2
2	The state-of-the-art	3
2.1	Interest point/region detection	3
2.2	Tracking	5
3	Stable Wave detector in 1D	6
3.1	The single scale Stable Wave detector in 1D	9
3.2	Multi-scale issues	9
4	The Stable Wave detector in 2D	10
4.1	2D Fourier transform	11
4.2	Single scale SWD algorithm in 2D	12
4.3	Multiscale SWD in 2D	12
5	The ultrafast low-level tracker	13
5.1	The concept of Zero Shift Points	14
5.2	The tracking algorithm	15
5.3	Implementation issues	15
5.4	Good points to track	16
6	The high-level tracker	17
6.1	Coarse to fine approach	18
6.2	Zero Shift Points (ZSPs) and the image pyramid versus the pyramid of ZSPs (and integral image)	19
6.3	Local coherence without an explicit geometric model of the target	19
6.4	Extensions of the locally coherent model	20
7	Results	23
7.1	1D robustness to noise	23
7.2	A simple single view experiment in 3D	24
7.3	Low level tracker	25
7.4	High level tracker	26
8	Conclusions	29
A	Resumé	35
B	Author's publications	36

1 Motivation and problem formulation

1.1 Motivation

The author of this thesis has been a member of the development team in a small Czech R&D company RS Dynamics, which designs and produces special and highly precise measuring instruments for the environment protection, security, etc. One of main products is the ECOPROBE series, which is dedicated to detection of traces of hydrocarbon contamination in the soil environment. The author has been in charge of the firmware and the electronics development for several different measuring instruments.

Let start with the original motivation of the research presented here. The ECOPROBE is used in outdoor measurements in places where the soil contamination occurs. It is necessary to log geographical coordinates of the location where the sample of earth gas was taken and analyzed. The differential GPS secures this task well in the cases when the satellites are visible. If the direction between the location of interest and satellites is occluded by the terrain or vegetation then GPS-based localization fails. This happens rather often in real-life measurements.

The initial idea was to use a stereo camera rig for the task known from mobile robotics as the simultaneous localization and mapping (SLAM). The thought is to capture sequence of images, establish correspondences among key points in them, estimate the movement of the observer and the 3D coordinates of the key points in the scene. The required precision was in the order of 0.1 meter in the area spanning several hundreds meters.

The usual approach is that several hundreds of key points are detected in the scene. Many methods for doing so were developed and have been used for this purpose. If the single typical representative has to be chosen then Harris corner detector [10] would pop out. It detects quite reliably points in the intensity image, in which strong partial derivatives in two perpendicular directions exist. These key points do not have any semantic meaning in the scene. Detected key points typically serve in a seek for a parametric transformation with relatively few parameters (say, up to 10). The problem is changed into a robust solution of an overdetermined system of equations usually solving a least square minimization task. As many more key points are available than necessary, the correct parameters are found quickly and reliably in most cases. Similar ideas are used in the case when key points are used in tracking in a videosequence.

When thinking about the described assignment in the context of a typical scene, in which ECOPROBE instrument is used, the idea of a Stable Wave emerged. The inspiration came from the nature. It is known that dogs are short-sighted and cannot see sharply. They still live well in the 3D world and are able to hunt. The thesis author is short-sighted too. He can also perceive approximately a 3D structure of the world without glasses. Another observation

concerns fast movements. If an object moves very fast then a human is unable to recognize details of it, however he still perceives a position and speed of the moving object. These observations motivated to seek some integral entities as blobs rather than differential structures as corners or edges.

This thesis explains the Stable Wave principle, suggests efficient algorithms to detect and refine it in images. The second key idea was to use the Stable Wave in tracking. The result is the detection/tracking system which is ultra fast and competitive with the state-of-the-art methods.

The thesis explicates the Stable Wave idea theoretically, suggests related algorithms, describes their implementation, and reports extensive experimental validation of the idea. In addition, the related C++ and MATLAB software is made publicly available to the research community. On the other hand, the author did not get the Stable Wave to the original application with ECOPROBE because he had to dedicate his efforts to other R&D activities in the company he works for.

1.2 Problem formulation

The research presented in the thesis aims at

- detecting *salient semantic-less regions of interest (stable waves)* in the image automatically. It is assumed that the input data come from a single camera videosequence which implies a short baseline;
- building *point correspondences* among detected stable waves in consequent frames of the videosequence;
- utilizing detected stable waves and established correspondences among them in a fast, *low-level tracking*.

The thesis builds on the key idea – the *stable wave*. The *wave is locally convex* and it is represented by a *unique point – the maximum*. The neighbourhood of this point constitutes the attraction basin leading to the mentioned point. This concept provides a competitive advantage as compared to the state-of-the-*interest points* approaches as *Harris points* [10], its simplified versions as FAST [27, 28] or its enhanced modifications as affine detector [21] or SHIFT [16]. The advantage comes from the *attraction basin* existence which does not require testing all points in the neighbourhood. The attraction basins can be observed with the evergreen *Kanade-Lucas-Tomasi tracker* [18] (abbreviated KLT in the literature) and in recent linear predictors trackers [11, 37].

We explicate the stable wave idea in one dimension (1D) first. The task of a *peak detection in 1D signals* was solved long ago and has many applications. The thesis offers a new alternative solution to the task which is fast, accurate, insensitive to noise, and computationally simple. In the thesis, the 1D

case serves as a point of departure to a two dimensional (2D) case needed in image/video analysis. Actually, the 1D stable wave idea is utilized in the RS Dynamics device for detecting explosives. Unfortunately, the real experimental data are company confidential and cannot be presented in the thesis.

The *tasks solved in the thesis* can be formulated in the three simple items:

- Explain the stable wave idea in 1D.
- Design the 2D version of the stable wave and study its properties.
- Use the 2D stable wave in tracking and evaluate its properties.

2 The state-of-the-art

The thesis deals, broadly speaking, with two areas:

1. *Detecting* of semantic-less salient entities – stable waves.
2. *Tracking* above mentioned stable waves in videosequences.

Both these areas have been extensively studied by others. Our intention is to keep the state-of-the-art description rather compact. We select mainly methods related to the key concept of the thesis, to the stable wave. The two next sections deal with detection (Section 2.1) and tracking (Section 2.2).

2.1 Interest point/region detection

The rationale of representing the image or part of it by a relatively *small number of semantic-less interest points/regions* is to provide reliable affirmations about image geometry and appearance. Sometimes these points/regions serve as basis for image description which is sparser and more informative than the image intensities (colours) itself. Points/regions are mainly used for matching slightly different images, stereo, 3D reconstruction, image retrieval, object categorization, object recognition, object tracking, etc.

Points/regions of interest have to be detected and many detectors were proposed for this purpose. The detected good points/regions should fulfil the following *four requirements* [7]:

1. *Robustness* against noise, image intensity variations due to lighting, geometric distortions, etc.
2. *Sparseness* to reduce the amount of data, keep relevant information and increase speed.
3. *High speed* because the detector/descriptors constitute low-level parts of more complicated algorithms.

4. *Completeness* because the detected points/regions should comprise much information about the image for subsequent processing.

The extensive survey of related detectors and methods is given in [35]. Detected entities are often divided into *corner detectors* (also interest point, key point), *blob detectors* and *region detectors*. Corners represented as points are essentially zero dimensional. Blobs and regions represent two dimensional entities. *Blobs* usually denote entities attached to a particular image location representing light or dark areas around it. *Regions* are explicitly determined by their boundaries.

To save space, detectors of one dimensional entities as edge detectors and line detectors will not be discussed here even they might be used for corner/region detection.

A popular group of corner detectors originated in *Moravec detector* [23], improved later by Zuniga [38]. The important breakthrough was the *Harris corner detector* [10] which is probably the most often used one also due to several easy to get public domain implementations.

A fast implementation of the Harris detector is described in [24]. Comparison of interest point detectors is given in [30]. An overview of existing interest point detectors can be found in [21].

Corners are not inherently scale invariant, i.e., a *multi-scale Harris detector* does not localize the same local structure at the same point in different scales. Mikolajczyk [21] choose the ‘correct’ scale as maxima of Laplacian-of-Gaussian in the scale space. Deriche [6] approached this problem by fitting the line through the locations at different scales and searched where the series of points converge. *Lowé’s detector* searches maxima of Difference-of-Gaussian [16] in the scale space. The recent thorough analysis of Harris corner detector is provided in [15].

It was observed by many that richer structures in images compared to local, ‘derivative-based’ corners can bring additional benefit. The extensions of Harris-like interest point detectors based on an affine normalization around Harris points were proposed [21], [29].

The other big group of methods relates to *region/blob detectors*. The seminal work from the same group as this work (the Centre for Machine Perception of the Czech Technical University in Prague) suggested *Maximally Stable Extremal Regions* (MSERs) [19]. It is based on an idea that thresholding the intensity image by all possible thresholds and observing the change of the area of detected regions allows to detect highly stable regions, those which correspond to the smallest speed of the area change. The approach was later extended by others, e.g. to color images [8].

The other regions detectors are, e.g., the edge-based region detector [33]; the detector based on intensity extrema [34], and the detector of ‘salient regions’ [12]. The performance of above region detectors is compared in [9], [22],

where the performance to change in viewpoint, scale, illumination, defocus and image compression are considered.

The *blob detector* proposed in this thesis was motivated by practical observations and a some practical experience with applying signal processing theory. Robustness and speed stems from basic properties of Fourier transformation, dot product and *phase of the first harmonics*. A global difference of phase of the first harmonics between two omnidirectional images was used to find their relative orientation [26]. However, the blob detection method suggested in the thesis uses a moving window focusing to individual landmarks in the image than to the whole image.

The blobs detected by method proposed here seek for local extrema similarly as [34, 19]. The difference of our method is in the requirement that the blob has to exhibit some degree of symmetry around its centre. The recent work of Maver [20] suggested a new interest point detector which explores symmetries. The approach uses much more complicated computational approach and is slower.

2.2 Tracking

The *aim of tracking* is to locate continuously the *target* in a videosequence. It is usually assumed that the target is given at the beginning of tracking. The *target trajectory* is built incrementally by repeatedly estimating relative displacement of the target between successive frames.

Many different tracking methods have been proposed, from global template-based trackers [1], shape-based methods, probabilistic models using mean-shift [4], particle filtering [14] to local key-point based trackers [25] or flow-based trackers [31].

The *original concern in this work* was in *tracking salient blobs* serving as natural landmarks in a videosequence captured in bush or forest-like scenes. The intended purpose was to self-localize the observer in the 3D environment. Similar task is approached in Simultaneous Localization and Mapping (SLAM) in mobile robotics [24, 3].

The term tracking covers a wide range of techniques which can be categorized according to a number of criteria: the character of the tracked object, e.g. a square image patch, a free-form region, an articulated structure or a dynamic texture, by the ability to adapt, learn and recover from failures, by the speed of tracking, by the choice of a predictor, and other properties.

This thesis focuses on *tracking image patches* and thus belongs to the category of *low-level tracking algorithms*. Two subtasks are integrated into the tracker: (1) the choice of objects to track and (2) the method of establishing correspondence between the instance of the objects in consecutive frames.

The *Kanade-Lucas-Tomasi (KLT) tracker* constitutes a milestone and the

widely used tool in low-level tracking. The tracker is based on the early work of Lucas and Kanade on image registration [18]. The idea was applied to tracking by Tomasi and Kanade [32]. Later the approach was explained clearly in the paper by Shi and Tomasi [31]. Additional information useful to understand the implementation were provided in [2].

KLT tracker establishes correspondence by (iterative) *gradient-based minimisation*. Typically, the tracked objects are image patches with two high eigenvalues of the structure tensor [31], which are essentially the same as regions centred around Harris interest points [10]. These so called ‘good features to track’ or Harris regions possess the property that they are different from all regions in their neighbourhood, a necessary condition for establishing reliable point-to-point correspondence. However, these points do not have, at least not by design, any property that would facilitate the minimisation step.

Other popular low-level tracking methods, e.g. [3, 13], rely on very fast detectors and establish correspondence by matching of detected points, which is feasible if only a small number of alternatives must be verified, i.e. when the error in prediction and hence the search radius is small with respect to the density of points.

Inspired by the active appearance models of Cootes et al. [5], Jurie and Dhome [11] realised that in some image regions an approximately linear relationship exists between observations and displacements, allowing ultra-fast tracking by performing a few dot-products in a high-dimensional spaces. This family of trackers is usually named linear predictor trackers.

The *linear tracker* has been recently generalised by K. Zimmermann et al. [37] in our Centre for Machine Perception at the Czech Technical University in Prague. They made the approach more precise by applying a sequence of lower dimensional linear operations which make progressively more accurate predictions. However, the significant weakness of linear prediction methods is the need, before tracking starts, to perform both a time-consuming search for suitable regions and learning of the linear mapping.

This work keeps the advantage of the linear predictor method – an extremely fast tracking, but removes the search for suitable regions to be tracked.

3 Stable Wave detector in 1D

The word ‘*wave*’ comes from Fourier transform which is a basic tool in signal/image processing. The term ‘*stable*’ says that the wave in the signal (or image) should be stable / well localized in some sense.

The *idea of the stable wave* originates from a simple experiment in 1D. The aim is to find a peak in 1D data reflecting measurements of a continuously varying physical quantity in equidistant discrete instants called samples.

We start from a data vector \mathbf{I} of length N . Suppose that the peak is approximately $T/2$ wide, where $T < N/2$, and it is located somewhere in the closed interval $\langle x_0, x_0 + T \rangle$, where $1 < x_0 < N - T$ (symbols $x_i \in \mathbb{R}$ measure the distance with subsample resolution from the beginning of data). This inequality allows the algorithm to move the window during its iterations. The reader may ask what the peak width is. The exact definition is not provided because it is difficult to establish it. For our purposes, the peak width is the half of period of the cosine which best fits the peak. The algorithm may *search the nearest peak of any polarity* or select the maximum or the minimum.

For the estimation of the phase, we compute the first harmonic in the form

$$A \cos\left(\frac{2\pi t}{T} - \varphi\right) = a \sin\frac{2\pi t}{T} + b \cos\frac{2\pi t}{T}, \quad (1)$$

where $a, b \in \mathbb{R}$ are sine and cosine coefficients of the first harmonic (computed by formulas for Fourier series), variable $t \in \mathbb{R}$ is relative to the position of the window (in contrast to x_i related to the beginning of data).

$$A = \sqrt{a^2 + b^2} \quad (2)$$

is the amplitude and φ is the unique angle in $(-\pi, \pi)$ satisfying (1), see Fig. 1.

Let $t_{\min}, t_{\max} \in \langle 0, T \rangle$. The first harmonic has the minimum and the maximum at

$$t_{\min} = \left(\frac{1}{2} + \frac{\varphi}{2\pi}\right) T, \quad t_{\max} = \begin{cases} \left(1 + \frac{\varphi}{2\pi}\right) T & \text{if } \varphi < 0, \\ \frac{\varphi}{2\pi} T & \text{if } \varphi \geq 0. \end{cases}$$

The peak position relative to the center of window is

$$\delta = \frac{T \arctan\left(\frac{a}{b}\right)}{2\pi} + d, \quad (3)$$

where the offset $d \in \{0, -T/2, +T/2\}$ depends on the signs of a, b and the desired task (the nearest peak/maximum/minimum). The coefficients a, b are computed from the window of length T , the same as the period of the base functions according to (5).

Let \mathbf{F} be a vector of intensity values in the window of length T . In the discrete case, sine and cosine are represented by vectors \mathbf{S} and \mathbf{C} , henceforth called the base vectors, which are defined as

$$\begin{aligned} \mathbf{S}(j) &= k_T \sin(2\pi((j + \tau)/T)), \quad \mathbf{C}(j) = k_T \cos(2\pi((j + \tau)/T)), \\ k_T &= \sqrt{T/2}, \quad j = 0, \dots, T - 1, \end{aligned} \quad (4)$$

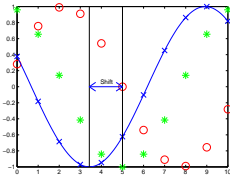
where k_T is a normalization constant and $\tau \in \langle -1, 1 \rangle$ tunes the position of extremes relative to the grid. The *Fourier coefficients* a and b are computed as

dot products

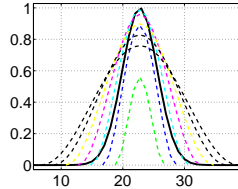
$$a = \sum_{j=1}^T \mathbf{F}(j) \mathbf{S}(j) = \mathbf{F} \cdot \mathbf{S}, \quad b = \sum_{j=1}^T \mathbf{F}(j) \mathbf{C}(j) = \mathbf{F} \cdot \mathbf{C}. \quad (5)$$

The peak can be localized by the following iterative algorithm:

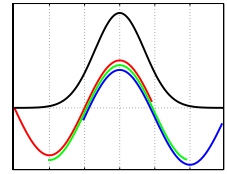
1. Define the window $\mathbf{W}_1 = \langle \hat{x}_0, \hat{x}_0 + T - 1 \rangle$, $\hat{x}_0 = \text{round}(x_0)$, $i = 1$.
2. Compute Fourier coefficients a_i, b_i from $\mathbf{F}_i = \mathbf{I}(\mathbf{W}_i)$ of the first harmonic wave (i.e., with a period T).
3. Estimate the peak shift δ_i in window \mathbf{W}_i according to (3).
4. $x_i = x_{i-1} + \delta_i$, $\hat{x}_i = \text{round}(x_i)$. If $\hat{x}_i = \hat{x}_{i-1}$ then finish. The peak location is $x_i + T/2$. If $|x_i - x_0| > T/2$ than fail – divergence.
5. $i = i + 1$. If $i > T$ then Fail – oscillation.
6. Define a new window $\mathbf{W}_i = \langle \hat{x}_{i-1}, \hat{x}_{i-1} + T - 1 \rangle$. Go to Step 2.



(a)



(b)



(c)

Obrazek 1: a) The sine (red) and the cosine (green) base functions. The arbitrarily shifted cosine wave (blue). b) Solid curves show the Gaussian $y = \exp(-(x - 22.76)^2 / 16)$. Dotted curves visualize the results of SWD at different periods T : 8 (green), 12 (blue), 16 (cyan), 20 (magenta), 24 (yellow), 28 (black), 32 (black). c) The principle of the CNW criterion. The minus sine wave (red, $a < 0$) corresponds to the rising slope. The minus cosine wave (green, $-b \gg |a|$) corresponds to the tip of the peak and sine wave (blue, $a > 0$) corresponds to the falling slope.

The algorithm typically converges in the first or the second iteration for the ideal peak without noise (e.g. Gaussian Fig. 1 b)) if the period T fits well the peak width. The algorithm works because the phase of the first harmonic wave determines the location of the peak. The *higher harmonics* make the peak shape more and more precise. Intuitively, the *symmetric peak* occurs in the location where the waves of different near frequencies meet with the same phase. If their phases differ then the peak becomes asymmetric, e.g. the rising slope is more steep than the falling slope. The algorithm works well for the period T more than one octave below or over the optimal period for isolated peak in flat area (Fig. 1 b)). The amplitude A (Equation (2)) can measure the strength of the response and suitability of the period T for a given peak.

3.1 The single scale Stable Wave detector in 1D

This section describes method *searching all peaks in discrete data*. Input vector \mathbf{I} of the length N containing data is covered by overlapping windows of the length $T < 2N$ which is the period of the base functions (4). The windows overlap by Ω samples. We found that T divisible by 4 and $\Omega = 3T/4$ allow fast computation – only $2N$ multiplications is necessary instead of

$$M(N, \Omega) = 2N \frac{T}{T - \Omega}, \quad (6)$$

in general case. There are special cases of $T = 4$ where no multiplication is necessary (the base vectors contain only $\{-1, 0, 1\}$) and $T = 8$ where just N multiplications are necessary (the base vectors contain $\{-1, 0, 1, \sqrt{2}\}$)

Following algorithm finds peak locations and SWD amplitudes, separately for maxima and minima:

1. Compute Fourier coefficients a_i, b_i of the first harmonic wave for individual windows \mathbf{W}_i .
2. Find the candidate windows using Consistent Neighboring Window (CNW) criterion (Fig. 1 c)).
3. For candidate windows, compute the shift δ_i (3) and amplitude A of the Stable Wave (2).
4. The extreme location relative to the candidate window \mathbf{W}_i is $\delta_i + T/2$.
5. Replace double detections of a single peak (a peak detected in two overlapping windows) by their mean.

The principle of CNW can be explained using Fig. 1 c): each minimum should follow the falling edge $a_i > 0$ and should be followed by the rising edge $a_i < 0$. Similarly, each maximum should follow a rising edge and should be followed by a falling edge.

3.2 Multi-scale issues

A *good peak of a certain width* produces *responses for several different SWD periods* as can be seen in Fig. 1 b). Notice, that the coefficients are computed always from the window of the length T , the same as the period of the base function. This is the reason for a seemingly impossible result – both algorithms mentioned above with the period T can detect the extremes of the sine or cosine wave of the period of $2T$ or even any longer. Also, the isolated peak much more narrow than $T/2$ in a flat area wider than T can be detected and precisely localized. This paragraph says that the algorithm works over the range specified at the beginning of Section 3. However, there are limits. Naturally, e.g. neither the cosine with period $2T$ cannot be localized by SWD with period T ,

similarly, nor the group repeating with period $2T$. The short period has also limitations, e.g. the rectangle T samples wide cannot be localized by SWD with period T or shorter.

Even though the SWD *algorithm does not require precise knowledge of the peak width*, its *estimate must be provided*. Such estimate may not be available in many practical situations or the peak width may vary in a wide range. To address this problem, the multiscale algorithm was suggested which searches a hierarchy of peaks at *several periods choosing the best scale* by some criterion (similarly to the other multiscale detectors [21], [16]).

Considering the excellent multiscale property of SWD, the scale steps can vary by an *octave*, i.e. form the series 2^k , $k = 1 \dots n$, where n is the number of scales. It is more sparse than 1.4^n used by Mikolajczyk [21] and Lowe [16]. The integer scale allows a more efficient computation compared to a non-integer scale.

The *pyramid* can be obtained using a variable period or a fixed (the shortest) period and data-pyramid obtained by subsampling the original signal and using an anti-aliasing filter. A suitable choice of the fixed period is 8.

Similarly to Mikolajczyk [21] or Lowe [16], the best scale can be chosen according to the strength of the response measured as the amplitude computed according to Equation 2. If the peak is near to the centre of the window then the amplitude may be replaced by the cosine coefficient b . Another criterion test the *multiscale stability* η estimated as

$$\eta = \min(|x_i - x_L|, |x_i - x_H|), \quad (7)$$

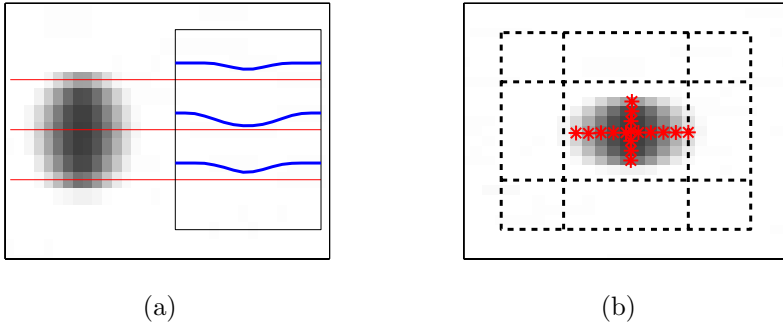
where x_i is the initial estimate at the period T_0 , x_L is estimated using the window of length T_L centred around x_i , and x_H is estimated using the window of length T_H centred around x_i . The stability can be evaluated at approximately a half of an octave lower (T_L) and higher (T_H) than the original scale (T_0), in which the peak was detected. The word ‘approximately’ means that the period is always an integer.

4 The Stable Wave detector in 2D

Let us observe the example of an *ideal blob in 2D* in Figure 2 similarly as we observed ideal peak in 1D. Assume an intensity profile along a line in the image domain intersecting a blob, e.g., a row or a column of the image. There is a peak on the 1D intensity profile.

The *peak in the intensity profile* along the line in the image is a necessary condition for the blob presence. This condition provides *two independent constraints* – one for rows and the second for columns. Assume that the SWD-1D algorithm from Section 3.1 is applied to the intensity profile along the image row. The SWD-1D localizes the peak, the representative point of the blob in

the row with subpixel precision, see Figure 2 a). These points lay approximately on a vertical line, shown as red stars in Fig. 2 b). Similarly, the results of SWD-1D on a column passing the blob form a horizontal line. The two lines are almost perpendicular. The *2D location of the blob* can be estimated as the intersection of these two lines also with a subpixel precision.



Obrázek 2: The image of blobs in a calibration pattern. a) The original image and the intensity profile along three rows intersecting the blob. b) The peak location of one blob from intensity profiles along rows and columns.

4.1 2D Fourier transform

A more thorough mathematical explanation of the suggested Stable Wave approach can be derived from the 2D Fourier Transformation (FT) and phase-based methods of stereo matching. The 2D FT is a vector function yielding the phase and the amplitude as the vector $[f_h, f_v]$ (horizontal and vertical frequency). Two independent frequency vectors are needed to obtain a 2D phase information. For a better imagination, the vector $\mathbf{f}_1 = [f, 0]$ corresponds to the wave parallel to the axis x . $\mathbf{f}_2 = [0, f]$ is the wave parallel to the axis y , and $\mathbf{f}_3 = [f, f]$ is the wave parallel to the line $y = 1 - x$.

Similarly to 1D case in Section 3, only a small part of Fourier transform coefficients will be used to localize a blob, i.e. just the coefficients and the phase corresponding to the two frequency vectors $\mathbf{f}_1 = [f, 0]$ and $\mathbf{f}_2 = [0, f]$. The period $T = 1/f$ will be the size of the window analogically as in Chapter 3.

Let us look in detail to the computation of the Fourier coefficient a for the vector \mathbf{f}_1 in the part (square window) of the image (matrix) \mathbf{I} . The base function $\mathbf{S}^{2D}(r, c) = \mathbf{S}(c)$ depends on c only. The sums can be decomposed as

$$\begin{aligned}
a &= \sum_r \sum_c \mathbf{S}^{2D}(r, c) \mathbf{I}(r, c) = \sum_c \mathbf{S}(c) \sum_r \mathbf{I}(r, c) = \sum_c \mathbf{S}(c) \mathbf{i}(c), \\
\mathbf{i}(c) &= \sum_r \mathbf{I}(r, c).
\end{aligned} \tag{8}$$

The meaning is the following. First, perform summation of the intensity of image $\mathbf{I}(\text{row}, \text{column})$ over the rows to get the vector $\mathbf{i}(\text{column})$. Second, transform the vector $\mathbf{i}(\text{column})$ by the 1D FT to get the coefficients and phase in the horizontal direction.

4.2 Single scale SWD algorithm in 2D

This section describes the *practical SWD algorithm in 2D*. The phases corresponding to frequency vectors $\mathbf{f}_1, \mathbf{f}_2$ can be used to localize the peak similarly as SWD-1D algorithm does. However this solution is not good. First, the application of the CNW criterion would be difficult. Second, the other problem is that the integrals for Fourier coefficients contain a large neighbourhood of the blob. The solid square in Figure 2a shows the optimal square window to detect the blob. The square areas in its corners bring just noise to the integrals.

Our *SWD-2D algorithm* combines the observation from Fig. 2 with the mathematical derivation. In short, we *detect peaks in rows and fit the vertical line v through them*. Next, we *detect peaks in columns* and fit the horizontal line h through them. The *location of the blob is estimated as the intersection of lines h and v* .

The algorithm works on gray-scale images (natural gray-scale or other scalar intensity image, e.g, a color component). Color RGB images can be converted to the gray-scale simply by summing their color components for each pixel.

4.3 Multiscale SWD in 2D

A multiscale hierarchy can be obtained by repeating algorithm described in Section 4.2 with different periods T . The series $T = 2^n, n \geq 3$ is a good compromise between the precision and the computational complexity. The other option is to downsample the image to get an image pyramid and run the algorithm on each image in the pyramid. The first option is a bit more precise, e.g. because the image pyramid is sensitive to rotation, etc. The second approach is more computationally efficient.

The simplest way of downsampling the image by the factor two is just taking every second pixel horizontally and vertically $\mathbf{I}_2(r, c) = \mathbf{I}_1(2r, 2c)$.

Such approach would suffer from *aliasing effects*. A filter should be applied. One possibility is to sum 3×3 neighbourhood around even pixels in both the even row and the even column to yield the intensity value for the pixel on a

more rough scale.

$$\mathbf{I}_2(r, c) = \sum_{i=-1}^1 \sum_{j=-1}^1 \mathbf{I}_1(2r + i, 2c + j).$$

Considering only *two scales it may be enough*. However, building the pyramid in such a way suffers from the following problem. The odd pixels in the first level propagate to the third level with higher weight than even pixels. The ratio is 1:2 for even-even:even-odd pixel and 1:4 for even-even:odd-odd. Therefore we use the following 3×3 filter

$$\mathbf{I}_2(r, c) = \sum_{i=-1}^1 \sum_{j=-1}^1 \mathbf{W}(2 + i, 2 + j) \mathbf{I}_1(2r + i, 2c + j), \quad \mathbf{W} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

This *filter propagates* each pixel with the same weight through the scales of the pyramid. Probably other filters may be found. However, the described filter seems to be a good choice. It is simple to implement and it yields a good performance. Another advantage of this filter is that its weights are powers of two. Also, the sum of its weights is $16 = 2^4$. Consequently, all computations can be performed in the integer arithmetics by simple bit shifts and additions which are core operations of all common processors.

5 The ultrafast low-level tracker

The low-level tracker establish *point to point correspondences* between *two consecutive images* in a video sequence or any other pair of images. The displacement between images must be small or predicted in advance with reasonable precision. The *low-level tracker* takes *two inputs*: (1) the *predicted location* of the key point, e.g. the key point detected in the previous frame of the sequence and (2) *the image* in which the point should be found which is the current frame of the sequence. The low-level tracker finds the new and a more precise position of the point in the current frame.

The suggested tracker works similarly as the Newton's method used in optimization. Having the initial pixel position, the tracker estimates the needed shift (the direction and the distance) to the key point and makes this shift. This step is iterated. The estimates become more and more precise as the algorithm approaches to the key point. The point \mathbf{x}_i in iteration i is considered as the key point if \mathbf{x}_i is closer to \mathbf{x}_{i-1} than some threshold. Typically as few as two to five iterations are necessary. At the key point, the *predicted shift is near to zero*, therefore the key points of the tracker are called *Zero-Shift-Points* (ZSP) in the rest of the chapter. If the initial estimate is precise enough then it is not necessary to consider and compare multiple tentative correspondences, the proposed algorithm just finds the correct one.

5.1 The concept of Zero Shift Points

Since ZSPs are defined in terms of local shift vectors Δ , we first explain their computation. The idea behind is given in Section 4.1. The mapping f which maps pixel $\mathbf{y} \in \mathbb{Z}^{2+}$ to shift vector $\Delta^+ \in \mathbb{R}^2$ estimates the position of the maximum of the first harmonic wave of the window centred at the pixel $\mathbf{y} = [r_0, c_0]$. Similarly, Δ^- estimates the position of the minimum. The elements of the shift vectors $\Delta^+ = [\delta_h^+, \delta_v^+]$ and $\Delta^- = [\delta_h^-, \delta_v^-]$ are computed according to Equation (9). The subscript ‘ \star ’ in δ_\star^+ , δ_\star^- , a_\star , b_\star is either ‘ h ’ (horizontal) or ‘ v ’ (vertical), respectively:

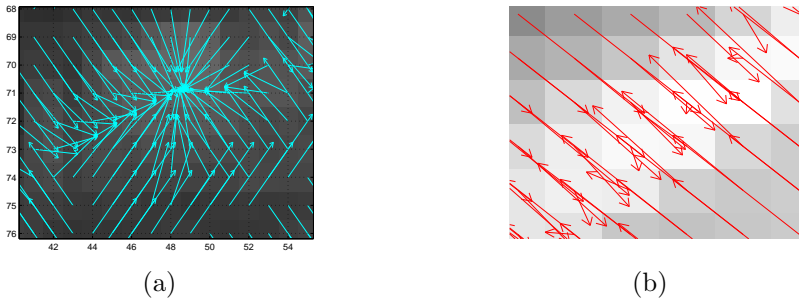
$$\begin{aligned} \delta_\star^+ &= \begin{cases} T \arctan(a_\star/b_\star)/(2\pi) & \text{if } (b < 0) \\ -T \text{sign}(a_\star)/4 & \text{otherwise} \end{cases} \\ \delta_\star^- &= \begin{cases} T \arctan(a_\star/b_\star)/(2\pi) & \text{if } (b > 0) \\ T \text{sign}(a_\star)/4 & \text{otherwise} \end{cases} \end{aligned} \quad (9)$$

where a_\star and b_\star stand for sine and cosine coefficients. They are computed according to Equation (10) from rectangular windows of the length T and width W centred at the point $\mathbf{y} = [r_0, c_0]$. Both T and W are odd integers. The best results were obtained when W was the nearest odd integer to $T/2$. The coefficients are defined as

$$\begin{aligned} a_h &= \sum_{i=-w}^w \sum_{j=-t}^t I(r_0 + i, c_0 + j) \mathbf{S}(j + t), \\ b_h &= \sum_{i=-w}^w \sum_{j=-t}^t I(r_0 + i, c_0 + j) \mathbf{C}(j + t), \\ a_v &= \sum_{i=-t}^t \sum_{j=-w}^w I(r_0 + i, c_0 + j) \mathbf{S}(i + t), \\ b_v &= \sum_{i=-t}^t \sum_{j=-w}^w I(r_0 + i, c_0 + j) \mathbf{C}(i + t), \\ w &= (W - 1)/2, \quad t = (T - 1)/2, \\ \mathbf{C}(i) &= \cos(\phi_i), \quad \mathbf{S}(i) = \sin(\phi_i), \\ \phi_i &= 2\pi(i + 0.5)/T, \quad i = 0, 1, \dots, T - 1. \end{aligned} \quad (10)$$

The point \mathbf{z} , in which Δ^+ or Δ^- becomes (approximately) the zero vector, is called a *zero shift point* (ZSP). There are two sets of ZSPs, ZSP^+ associated with maxima (and the Δ^+ field) and ZSP^- with minima (and Δ^-) of the intensity function, respectively. Since the processing of the two sets is identical and independent, we do drop the superscripts in the sequel. Such points are subpixel entities, i.e. they rarely appear at pixel centres. Vectors Δ at pixels around \mathbf{z} are pointing close to \mathbf{z} as depicted in Figure 3.

Typical *locations of ZSPs are centres of approximately elliptical ‘blobs’*, i.e. symmetric areas with higher/lower intensity than their neighbourhood (Figure 3a), narrow edge features (Figure 3b) and flat areas which are bigger than the length T of the moving window. ZSPs on ridge features and flat areas are not suitable for tracking since they are unstable and can be filtered out by simple rules, see Section 5.3.



Obrázek 3: Examples of typical shift fields (a) near a single ZSPs in a blob-like region and (b) near a ridge with multiple ZSPs.

5.2 The tracking algorithm

The following simple iterative algorithm tracks a single ZSP from its position $\mathbf{x}_0 \in \mathbb{R}^2$ the previous frame (or from a prediction of its position in the current frame) to a subpixel position $\mathbf{x} \in \mathbb{R}^2$ in the current image.

1. $i = 1$, $\mathbf{y}_0 = \text{round}(\mathbf{x}_0)$
2. Test the closeness of \mathbf{y}_{i-1} to image margins. If \mathbf{y}_{i-1} is more near to the border than $T/2 + 1$ then fail.
3. Compute the shift vector Δ^* in pixel \mathbf{y}_{i-1} according to Equation (9). The superscript ‘ \star ’ in Δ^* is either ‘+’ if the minimum is tracked or ‘-’ if the maximum if tracked. The new position is $\mathbf{x}_i = \mathbf{y}_{i-1} + \Delta^*$.
4. Test the convergence. If $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_\infty < z$ then $\mathbf{x} = \mathbf{x}_i$. Finish.
5. Test the divergence. If $\|\mathbf{x}_i - \mathbf{y}_0\|_\infty > d_m$ then fail.
6. $i = i + 1$, if $i > n$ then fail.
7. $\mathbf{y}_i = \text{round}(\mathbf{x}_i)$. Go to step 2

5.3 Implementation issues

The use of *integral images* significantly speeds up the tracking of a large number of points. The integral images are cumulative sums along image rows and columns:

$$\begin{aligned} J^h(r, 0) &= I(r, 0), & J^h(r, c + 1) &= J^h(r, c) + I(r, c + 1), \\ J^v(0, c) &= I(0, c), & J^h(r + 1, c) &= J^h(r, c) + I(r + 1, c). \end{aligned} \quad (11)$$

The *2D convolutions* for sine and cosine coefficient evaluation (see Equation 10) can be efficiently computed if the intensity values are first summed along the width of the window to get vector \mathbf{V} of length T . Then, the coefficients are computed as the dot products of the base vector \mathbf{S} or \mathbf{C} with \mathbf{V} . By using

integral images, the sum of W values for each element of \mathbf{V} can be replaced by a single subtraction. The coefficients related to the pixel $[r_0, c_0]$, $r_0 > T/2$, $r_0 < m - T/2$, $c_0 > T/2$, $c_0 < n - T/2$, can be efficiently computed as follows:

$$\begin{aligned}
\mathbf{V}^h(i+t) &= \sum_{j=-w}^w I(r_0+j, c_0+i) = \\
&= J^v(r_0+w, c_0+i) - J^v(r_0-w-1, c_0+i), \\
\mathbf{V}^v(i+t) &= \sum_{j=-w}^w I(r_0+i, c_0+j) = \\
&= J^h(r_0+i, c_0+w) - J^h(r_0+i, c_0-w-1), \\
w &= (W-1)/2, \quad t = (T-1)/2, \quad i = -T/2 \dots T/2, \\
a_h &= \mathbf{S} \cdot \mathbf{V}^h, b_h = \mathbf{C} \cdot \mathbf{V}^h, a_v = \mathbf{S} \cdot \mathbf{V}^v, b_v = \mathbf{C} \cdot \mathbf{V}^v.
\end{aligned} \tag{12}$$

The computational cost of Algorithm ?? depends mainly on the following three operations:

1. *Preprocessing*, i.e., the integral image calculation: requires just two integer additions per pixel.
2. *Calculation of the coefficients*: requires T integer subtractions and T integer multiply-accumulate instruction per coefficient; four coefficients are necessary per each iteration of a single point.
3. *Computation of the shift vector Δ* : requires 2 floating point division, 2 floating point arctan operations and 2 floating point multiplications per each iteration of a single point. The low level tracker needs about 4 iterations for each point.

The low level tracker spends less than $10\mu\text{s}$ on each point per frame on the HP6540b notebook. The implementation runs in a single thread and some parts of the code are still far from being optimal.

5.4 Good points to track

A good point to track should be *unique in its neighbourhood* and the tracking algorithm should be able to follow it over a wide range of disturbances. There are ‘simple’ disturbances like translation, rotation and scale change. In these cases, it is possible to identify easily ZSPs, which will be sufficiently robust, just by testing the original image where the points were detected. Some ZSPs are resistant to the affine transform distortion up to some extent (when the ellipse becomes too elongated). The perspective distortion destroys the symmetry. Thus, it always reduces the precision of the estimated point location. If the perspective distortion is more severe then it causes disappearance of the ZSP.

The *best point to track* by algorithm from Section 5.2 is the *centre of an ideal blob* like the one shown in Figure 2. The *localization* of such point is completely rotation invariant and robust to the scale and affine change in a wide range. In

real images, there are almost no ideal blobs. However, some patches are similar to them and can act as good targets for tracking.

On the other hand, there are some *clearly unsuitable ZSPs, which can be easily discarded*. These bad ZSPs are in flat areas (one or both of b_* coefficients are small) and on ridge features (Figure 3b). The unstable ZSPs $[r, c]$ on ridge features can be detect using criterion

$$|\delta_v(r, c \pm t)| > kt \quad \text{or} \quad |\delta_h(r \pm t, c)| > kt, \quad t = \text{ceil}(T/8), k < 1, \quad (13)$$

where k is a tuning parameter (default 0.8).

The quality of ZPS \mathbf{x}_0 detected at period T can be measured by its rank obtained by testing the presence of ZSP \mathbf{x}_+ and \mathbf{x}_- at the same polarity with periods T_+ (the nearest odd integer to $T+T/4$) and T_- (the nearest odd integer to $T - T/4$), respectively. ZPS of rank 0 are detected only with period T . ZPS \mathbf{x}_0 has rank 1 iff only one of \mathbf{x}_+ , \mathbf{x}_- exists more near than $T/4$. ZPS \mathbf{x}_0 has rank 2 iff both \mathbf{x}_+ , \mathbf{x}_- exists more near than $T/4$.

Good blobs to track are detected as ZSPs for periods in a wide range around of the optimal period. The locations of ZSPs localizing such blobs vary only slightly with the change of the period. This is important for two reasons. First, when the scale between consecutive images changes then the point can be localized using the same period with a small error. Second, when searching good ZSPs in the first image (or new ZSPs in a changing scene) then it is not necessary to scan all possible periods (all odd numbers from 5 to approximately a quarter of the size of the image) but only some selected periods in the exponential series (e.g., the series 9-19-39-79-159 was used in experiments of this chapter for typical images 640×480).

It is not necessary to test each pixel for the zero-shift condition since ZSPs suitable for tracking lie inside a basin of attraction with the size approximately equal to $T/2$ or bigger. Therefore it is enough to start tracking (by algorithm from Section 5.2) from points on a regular grid with a period of $\approx T$.

6 The high-level tracker

This section deals with the *integration of the low level tracker* from Section 5 into a *tracking system* aiming to track a more complex object than mutually independent points. One of the desired applications is to provide data for the camera pose estimation or/and for the mapping of the environment in which a human or a robot is moving. The high level tracker is called simply tracker inside this section.

The general *problems solved by a tracker* are:

1. To discriminate the target against background.
2. To estimate a global motion.

3. To detect malfunctions of the tracking.
4. (Optionally) to predict a global motion in the next frame.

In this work, the *target is represented as* a cloud of trackable points (or regions) having a similar (model dependent) behaviour. Tracking the target means to track these points and to estimate the global motion from individual local motions of individual points. Such *tracker must solve the following problems*:

1. Tradeoff between the range and precision of tracked points.
2. Prediction of the local movement of individual points.
3. Detect outliers – mismatches.
4. Handle lost points – no corresponding point found in new image which is detected at the lower level.
5. Estimation of the global motion from the individual local motions of tracked points.
6. Short range of available ZSPs (Zero Shift Points) when tracking a small and fast target.

6.1 Coarse to fine approach

The crucial part of the tracking is to solve the *tradeoff between the range and precision of the tracked points at different scales* properly. This problem is common to all trackers based on point/region correspondences [37, 3]. This problem is solved in the low level tracker sometimes, e.g. in the pyramidal KLT tracker in OpenCV library [36, 2]. The points corresponding to bigger regions (a coarser scale) have a longer range and they can handle a longer displacement between consecutive images. However, the localization precision is worse compared to points corresponding to smaller regions (a finer scale). And vice versa, the points from the finer scale are more precise, however, they have the shorter range. The scale corresponds to period T in the case of Zero Shift Points (ZSPs).

Generally, the system should work in *coarse to fine manner*. At coarse scales, the rough estimates of movement should be made and this information should be propagated into finer scales to achieve good precision of resulting global model (between frame homography or 3D pose estimation and environment map). The proper *propagation of information from coarse scales to fine scales* is essential. It is necessary conditions for tracking the points corresponding to small regions (which have small range). It is crucial to identify outliers at each scale to avoid propagated mismatches at lower levels which may cause the resulting global model completely invalid. The point to track have no chance to correct bigger prediction error than its range. The low level tracker may just say ‘I am lost’ by the comparison of surrounding regions in the current image

and the previous image or if the tracking distance (from initial position) is too long.

6.2 Zero Shift Points (ZSPs) and the image pyramid versus the pyramid of ZSPs (and integral image)

In a coarse to fine approach, there are two ways how to handle the scale. The first possibility is to build an image pyramid and use regions of the same size (e.g. the same smoothing σ for Harris detector or the same $N \times N$ mask for KLT tracker [2]). The second possibility is to work with the original image and change the size of regions. The first approach includes a costly preprocessing (an anti-aliasing filter should be applied) bringing the advantage of a scale independent complexity when tracking individual points. The second approach does not require the preprocessing step, however, the complexity of tracking each point depends (usually linearly) on the area of the region (i.e. it typically depends quadratically on the range).

In the case of ZSPs, a very simple integral image (see Section 5.3) can be used to reduce the complexity while tracking each point. The computation of this integral image is much faster than building the image pyramid. Using the integral image, the complexity of tracking each point depends linearly on its range. Next, the number of ZSPs with period T decreases quadratically with growing T . As a result, the time spent on a more coarse level of the point pyramid is shorter than the time spent on a more fine level. Changing the period is more flexible/cheaper than changing the scales of individual levels of image pyramid. It also allows to follow the scale change of target more precisely and independently if several target are tracked in a single image. Therefore, the point pyramid instead of the image pyramid will be used in the rest of this chapter, even though both approaches are possible when ZSPs are used. The pyramid which levels differ in the period by the octave is used.

6.3 Local coherence without an explicit geometric model of the target

This model imposes probably *minimal assumptions on the target and its movement*. Only a *local coherence* is assumed as close points move similarly. The local movement is approximated by the translation. Small scale changes, rotation, and perspective deformations of the target can be approximated by the different local translation in different parts of the target. The global motion is not evaluated by the tracker itself. The target is a cloud of points covering either the whole reference (first) image or its part. The *locally coherent model*:

1. Groups ZSPs into the levels of pyramid.

2. Describes the neighborhood of each ZSP \mathbf{x}_i on its level of pyramid, i.e. finds the set of m nearest points or the set of m_i points which are closer than some maximal distance. The choice of the neighborhood set may depend on the application. In the experiments presented, the m nearest points are preferred to guarantee enough number of points for the outlier identification.
3. Assigns the predictor to ZSPs on all pyramid levels except of the highest level. The predictor of ZSP at the level l is the nearest ZSP at the level $l + 1$.

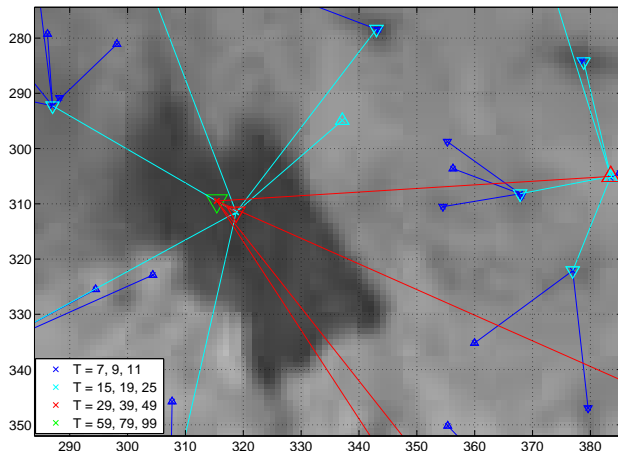
The locally coherent model is *tracked* by the following algorithm

1. Set level l to the highest level of the pyramid.
2. Track individual ZSPs at the level l .
3. Detect outliers – points whose disparity differs from the disparity of neighbors more than δ_m .
4. (Optional.) Remove points that are outliers for more than k consecutive frames. The neighborhood sets and predictors should be rearranged.
5. Correct the outlier disparity – replace the disparity of outliers by median/mean disparity of their neighbours.
6. Correct outlier position for the next frame to its initial position plus disparity.
7. (Optional.) Retrack outliers from their corrected positions. Consider ZSPs shifted more than δ_m as outliers and return them back to their position before retracking.
8. If $l = 1$ then Finish.
9. $l = l - 1$. Propagate the disparity to ZSPs on level l from $l + 1$ – add disparity of the nearest point from the higher level to the initial position of the ZSP to be tracked. Go to Step 2.

Fig. 4 shows the locally coherent model on the detail of the first image of ‘Mouse pad’ sequence. There are blobs of different size and quality. Most of them are small blobs detected just on the lowest level. Bigger blobs are usually (not always) detected on several levels (2-3) it means they are robust to scale change of more than 3-4 octaves. *Well symmetric blobs* (e.g. eyes – two most top-right cyan/blue triangles or the most top-right group of red, cyan, blue triangles) are localized in almost the single point at several levels.

6.4 Extensions of the locally coherent model

The *basic locally coherent model* may be *insufficient for long tracks*, in which the change of the global scale (moving target) or the local scales (moving camera – the near regions change their scale more fast than more far regions) is



Obrázek 4: A locally coherent model illustrated on the detail of the 'Mouse pad' sequence. The color distinguishes the levels of the pyramid (blue – lowest, cyan, red, green – highest). Crosses show locations of ZSPs, the orientation of triangles (the same color/location as the cross) shows the polarity of the wave, more precisely of the cosine coefficient. The line of the same color as the cross/triangle connects ZSP at level l to its predictor ZSP at the level $l + 1$.

significant. Therefore, this tracker may be considered more as a middle-level than a high-level tracker (equivalent approximately to the pyramidal KLT tracker in the OpenCV library [36, 2]).

The *performance can be improved* if the *tracker cooperates with the application*, which uses the tracked points. It means if the tracker can use information revealed by the application such as global or local scale, prediction of global movement or further outlier detection criterion and outlier correction mechanism. Probably the most important information is the *scale estimate for the period adjustment* of the low-level tracker.

6.4.1 Locally coherent model and RANSAC estimating planar homography

The connection of the tracker with the locally coherent model to RANSAC estimating homography is used as a model example how to create the application exploring the advantages of tracking ZSPs. Suggested system tracks the planar target and estimates homography between the reference (e.g. the first) frame and the current frame. The *homography* may be used for *several purposes* inside the tracker:

1. The *estimate of the scale change for the period adjustment*.

2. *Outlier detection* according to reprojection error.
3. *Refreshment*. The points from the reference image may be time to time (e.g. each 5th – 100th frame) projected by homography to the current image, retracked to eliminate imprecision of homography and to adopt to possible changes of target and then used to track in the new image instead of possibly lost or drifted points tracked frame by frame. After retracking, the points that moved too far from their projected positions should be removed (or alternatively their position may be adjusted in the reference image by the inverse of the homography).
4. *Renew*. The deformation of the targeted or significant scale or perspective change may cause ZSPs (which are continuously tracked from the reference image) to disappear or become unstable. If such problem is detected or expected (e.g. scale changed significantly) it is better to set the current image as reference, find a new set of ZSP and construct new model. Next tracking will be performed with new model and homographies will be estimated between new reference image and current image. If the homography between the first and the current image is necessary then it can be computed by multiplying the homography matrices of individual subsequences. It is not recommended to perform renew too often, because each renew potentially introduce drift due to the break of feedback between the current and the first image formed by point correspondences. Each estimated homography has some imprecisions and they are accumulated.
5. *Warp*. The new image may be warped using homography between reference and last tracked image to minimize geometric distortion. It eliminates the need for period adjustment and reduce effect of perspective on ZSP localization. It seemingly saves ZSPs with short periods from disappearing, however, blob formed from single pixel by warp may be very unstable. Aliasing effects may occur when ‘zooming out’ by warp.

6.4.2 Range extension

Small target may contain only small amount of ZSPs with long period and even their periods may be too short to have sufficient range for fastly moving target or fast rotation around principal axis. Both the *range and resistance to rotation can be easily extended* using approach which generate and test hypothesis about global motion. Algorithm ?? describes generally The range any tracker \mathbf{T} which is able to evaluate the quality of tracking result can be extended by the following algorithm:

1. Track points $\{x_i^0\}$ using tracker \mathbf{T} to $\{x_i^1\}$.
2. If tracking quality is sufficient then finish, $\{x_i^1\}$ are tracked points.
3. For all available global motion hypothesis \mathbf{H}

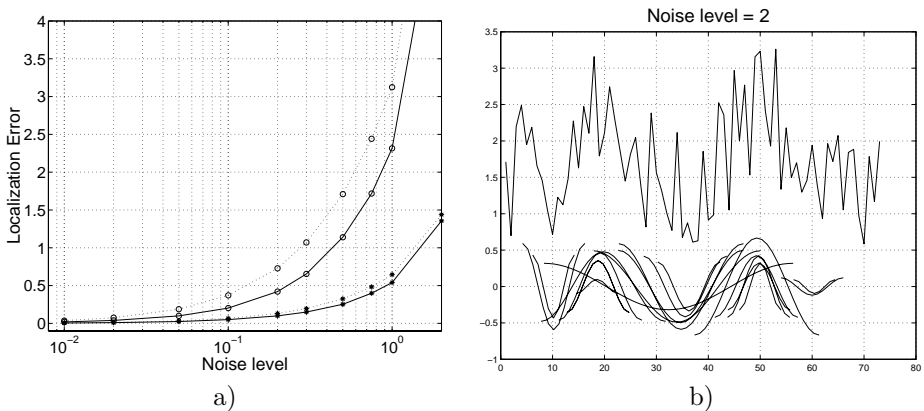
- a) Transform $\{x_i^0\}$ to $\{x_i^{0H}\}$ according to \mathbf{H} .
 - b) Track points $\{x_i^{0H}\}$ using tracker \mathbf{T} to $\{x_i^{1H}\}$. If tracking quality is sufficient then finish, $\{x_i^{1H}\}$ are tracked points.
4. If all available hypothesis \mathbf{H} were tested then select $\{x_i^{1H}\}$ with the highest quality or $\{x_i^1\}$ if its quality is better than all $\{x_i^{1H}\}$

There are obvious global motion hypotheses:

1. Rotation around the principal axis.
2. Translation.
3. The combinations of the two previous.

7 Results

7.1 1D robustness to noise



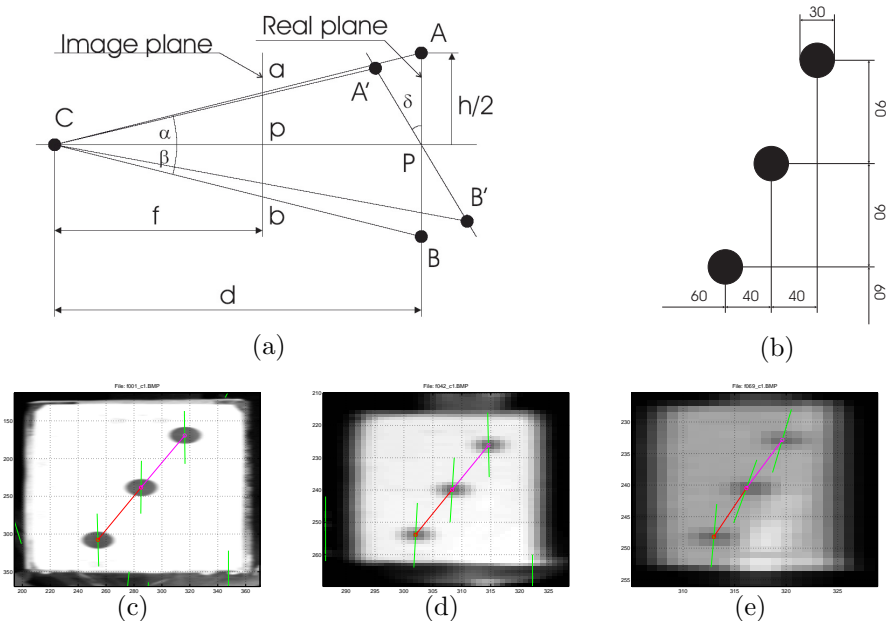
Obrázek 5: Robustness to noise. The results are from 100 repeats. The uniform noise from the interval $\langle 0, \nu \rangle$ was added to the signal containing two Gaussian peaks of the unit amplitude. The signal was vertically shifted by adding the value 0.5 just for visualization purposes. a) Error versus noise level ν . b) . . . f) Examples of data of $\nu = 2$.

The robustness of the multiscale SWD algorithm to noise was tested on synthetic data. The signal containing two Gaussian peaks

$$\exp\left(\frac{-(x - 52.76)^2}{51.2}\right) \quad \text{and} \quad \exp\left(\frac{-(x - 17)^2}{12.8}\right),$$

$x \in \{1 \dots 81\}$ was degraded by the additive uniform noise. The uniform distribution was used as a good model of the quantization noise. Independent instances of uniform noise vectors $\Psi_i, i = 1 \dots 100, \Psi_i(j) \in [0, 1]$ were generated, scaled by the noise level ν and added to the signal S . The results are summarized in Figure 5a). The standard deviation of the estimate was smaller than one sample up to the noise level $\nu = 1$.

7.2 A simple single view experiment in 3D



Obrázek 6: a) The setup of the distance measurement experiment from a single view. b) The target pattern used in the experiment. The image (enlarged details) of the target observed from different distances c) 1m, d) 5m, e) 9m.

The aim of this experiment was to demonstrate usefulness of SWD in a simple navigation task. Namely, the goal was to measure a distance to an artificial label observed by a cheap camera. A very simple measurement method together with a simple experimental setup (Fig. 6) were chosen deliberately in order to minimize unknowns and to find ground-truth easily.

Table 1 summarizes results of the experiment. The distance between markers measured in the image ranged from 75 pixels for $d = 1\text{m}$ to 7.6 pixels for $d = 10\text{m}$. The change in the image distance between the markers for $d > 8\text{m}$

Real distance[m]	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
Mean est. dist.[m]	1.004	1.981	2.983	3.992	4.973	5.991	6.944	8.042	9.065	9.954
Mean abs. error [mm]	9.6	20.2	17.3	16.1	35.9	37.2	70.0	54.2	116.0	112.5
Mean abs. error [%]	1.0	1.0	0.6	0.4	0.7	0.6	1.0	0.7	1.3	1.1
Max. abs. error [mm]	32.6	39.2	36.8	40.2	82.2	91.6	207.5	199.8	364.4	318.5
Max abs. error [%]	3.3	2.0	1.2	1.0	1.6	1.5	3.0	2.5	4.0	3.2
Marker size [pix]	24.0	11.0	7.0	6.0	6.0	5.0	4.0	4.0	3.0	3.0
Marker distance [pix]	75.4	38.2	25.4	18.9	15.2	12.6	10.9	9.4	8.3	7.6

Tabulka 1: The error of the measurement of the distance between the camera and the target plane.

is less than 1 pixel. However, the localization error is still better than 6cm at $d = 8m$. It corresponds to 0.06 pixel precision in measurement of the marker-to-marker distance.

7.3 Low level tracker

The experiments presented below focus mainly on the performance of the low level tracker and examine its properties necessary for its successful integration into the higher level tracking algorithm. The most important properties are the range of the tracker (the maximal inter-frame disparity or the predictor error which the tracker can handle), the ability of the tracked ZSPs to survive photometric and geometric changes, and the precision of tracking. The experiments use publicly available real sequences with available ground truth data [37].

The *attraction basin* of ZSP \mathbf{x} is a set of pixels from which the tracking algorithm reaches \mathbf{x} with some tolerance θ . Fig. 7 a), b) shows the attraction basin of ZSP with polarity ‘+’ and rank 2 grouped according to the period of ZSP. Notice that the basins are not symmetric and ZSP is usually located far from the centre. The density of ZSPs with ‘-’ polarity and their basins are similar (not shown). Figure 7 shows the shapes of cumulative attraction basins (a, c, e) and the estimate of the tracker range (b, d, f) in the first image of the ‘Mouse pad’ sequence.

The cumulative attraction basins were observed on various images with similar results: about 50% of ZSPs has the range at least half of their period.

Extensive experiments on synthetic images generated by warping real image by a known transformation were performed. These experiment proved rotational invariance and good robustness to scale changes, affine and perspective transformation.

The guided track experiment evaluated the repeatability of Zero Shift Points (ZSPs) on real sequences. More precisely, the experiment tests the ability of

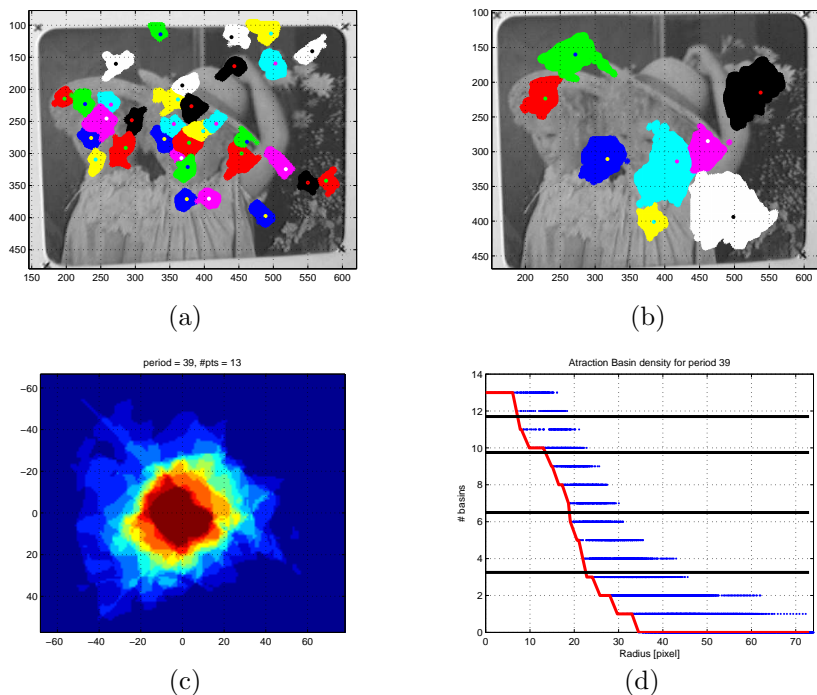
ZSPs to survive a geometric distortion together with other disturbances present during the video sequence capture (e.g. blur, noise, illumination changes and JPEG compression). Except of geometry, the other distortions were not quantified either controlled during capture of the sequences. The repeatability evaluated by the experiments in this section measures the presence of a ZSP near the expected location not the range of displacement the tracker can handle. It proved good endurance of ZSPs. The period adjustment improved significantly the repeatability. ZSPs with period 19 and longer (in the first image) had repeatability over 80% in all frames of the sequence. In most of the frames the repeatability of ZSPs with long period was near 100%. Smaller blobs suffered by blur as any other innterest points on small scale would do.

7.4 High level tracker

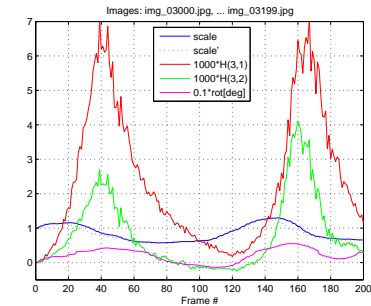
Fig. 8 shows performance of the locally coherent model on the part of ‘Mouse pad’ sequence with big changes of geometric distortion. The tracker with and without RANSAC is compared.

The tracker with the improved locally coherent model (ILCM) was compared with the state of the art reported by [37], the same evaluation method was used. ILCM tracker uses homography estimates by RANSAC to refresh every 5 frames and to renew if the scale changed out of range $\langle 0.6, 1.5 \rangle$ or too many points were lost. The scale change for period adjustment was estimated independently on homography. The robustness measured by the numbers of loss-of-locks on ‘Mouse pad’ sequence are 13 (NoSLLiP, [37]), 281 (SIFT, [17]), 398 (KLT, [18]), 1083 (SLLip, [11]), 93 (SLLip, half range, [11]). The ILCM tracker had 92 loss-of-locks. The number of frames in the sequence was 6935. The numbers of loss-of-locks on the ‘Towel’ sequence (3229 frames) was 5 (NoSLLiP) versus 8 (ILCM). The ‘Phone’ sequence contains 2300 frames provided on the web. However the results published in [37] are from the first 1800. The numbers of loss-of-locks on the first 1800 frames was was 20 (NoSLLiP) versus 9 (ILCM). The last 500 frames are the most difficult ones: ILCM tracker has another 8 loss-of-locks there, i.e. 17 in all 2300 frames. The reasons for loss-of-locks of ILCM tracker was the same in all three sequences: the fast movement of the small target.

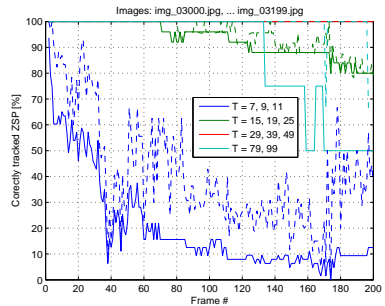
The ILCM tracker with range extension had significantly better performance – the numbers of loss-of-locks was 11 on ‘Mouse pad’ sequence, 5 on ‘Phone’ sequence and 6 on ‘Towel’ sequence.



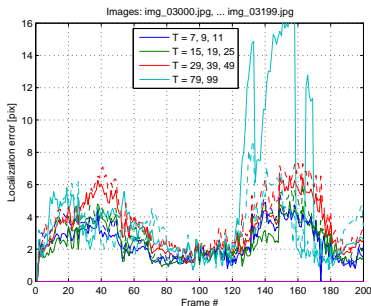
Obrázek 7: Attraction basins in the first image of the 'Mouse pad' sequence. ZSPs found with period a) 19, b) 39. Polarity of ZSPs is +. The dot of different color inside of each basin marks location of ZSP. c) Cumulative attraction basins in the first image of the 'Mouse pad' sequence. d) Each blue point corresponds to a pixel in the images of cumulative attraction basin and depicts the relation between the radius from ZSP and the number of basins covering the pixel. Black lines show 25%, 50%, 75% and 90% of the total number of basins. The red curve predicts the minimum number of correct correspondences found by the low-level tracker for a displacement of a given radius in an arbitrary direction.



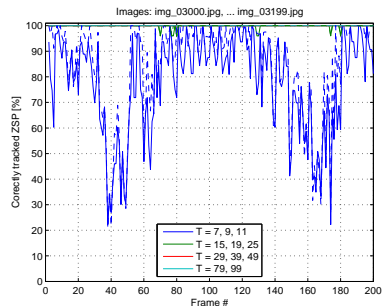
(a)



(b)



(c)



(d)

Obrázek 8: The 'Mouse pad' sequence. Tracking with the model of the local coherence with the period adjustment. a) Geometric distortion parameters (current to reference image) estimated from the ground truth and the scale (marked with ' in the legend) estimated from the tracked point. b) The percentage, of correctly tracked points consistent with the ground truth (without RANSAC). c) The localization error of inliers (identified by the tracker itself), solid lines – average error, dashed lines – 80% percentile, d) The percentage, of correctly tracked points consistent with the ground truth (with RANSAC). Solid lines – all points consistent with the ground truth, i.e. both the points which the tracker recognized as inliers and the outliers which the tracker corrected. The correct position is evaluated for the latter points.

8 Conclusions

The main contributions (sorted by their importance) of the work presented are:

1. The *ultra fast low level tracker* was proposed and described. The ‘low level’ means that the tracker establishes point-to-point correspondences between two consecutive frames of the videosequence. The local patches surrounding the points to be tracked are roughly rotationally symmetric blobs which are brighter or darker than their background. The tracking time of a single point is in the order of microseconds.
2. The *criteria to select good blobs* to track and the fast method to search them were explicated in Section 5.4. We demonstrated the long endurance of these points in tracking on real sequences, their repeatability and precision on synthetic data.
3. The *Zero Shift Point (ZSP)* pyramid was proposed, i.e. the pyramid grouping the points depending on their scale for the multiscale tracking on a higher level of abstraction (tracking not individual points but a more complex pattern connecting the points). A simple integral image idea is used while working with the ZSP pyramid and replaces the traditional image pyramid.
4. The *locally coherent model* was designed which imposes only a weak assumption on the target and its movement, i.e. that near points move similarly. The local motion is approximated by a pure translation. It was demonstrated that this simple model is able to cope even with relatively big scale changes and perspective deformations on real data in Section 6.3. The tracker based on a locally coherent model is able to identify and correct outliers of the low level tracker (which are probably incorrect matches). The tracker self-evaluates the quality of tracking based on the number of outliers.
5. *Examples were provided how to use the tracker* with the locally coherent model in the application using point-to-point correspondences established by the tracker. The testing application was the RANSAC procedure estimating homography between the current and reference image. It was showed how to use the information gathered by application as a feedback to the tracker to significantly improve the performance of the whole system.

The *method extending the range of movements* that the tracker can handle was suggested. The method is based on the generation and test of global motion hypotheses and on the ability of the tracker to self-evaluate the quality of tracking. This method is not applicable in general for combinatorial complexity reasons as it is dependent on the number of hypotheses.

However, the method is useful in the case of the tracker with the locally coherent model. In such a case the tracker is able to self-detect a failure and ask for a new hypothesis. The failures occur rarely and 14 basic hypotheses significantly reduced the number of loss-of-locks. It costs 2.3 times more in average if compared to the situation when this mechanism is not used.

The method *solves the problem of a small fast target* which hurts all trackers utilizing the attraction basin as KLT or linear predictors. The trackers which detect and match have this method built in naturally in some limited extend, in the limited neighbourhood around the expected position, in which all key points are considered as tentative correspondences.

Less important contributions:

1. The *Stable Wave Detector* (SWD) was designed for a wide baseline or the detect-and-match approach. The experiments showed the subpixel precision in the localization of ‘good’ blobs, e.g. for the camera pose estimation or for designing fiducial markers for ground truth computations.

The circle as the fiducial marker seems better than crosses. A colleague from our department Karel Zimmermann used crosses as fiducial markers while calculating ground truth on a ‘Phone’ and ‘Mouse pad’ sequences. It was observed that crosses suffer by decreasing size and blur while circles of the same size did not manifest this unwanted behaviour. The crosses after decreasing resolution and blur look more like circles. The corner detector seeking ground truth locations likely does not find what it is designed for. The outcome is an error of several pixels in the calculated ground truth. The experiments with SWD have shown that this undesirable effect is much less pronounced with circles.

2. *Peak detector* was proposed, i.e. SWD in 1D. Both SWD blob detector and the ultra fast low level tracker are based on its general idea of the localization based on the phase of the first harmonic wave. We did not do any comparison to other peak detectors, however due to its precision, robustness to noise and a very low computation complexity it may be ideal for practical application. The peak detector was already used in a hand held explosives analyzer produced by RS Dynamics. Here the low power consumption was important issue. The details cannot be mentioned here because of company confidentiality.

Reference

- [1] S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261—271, February 2007.
- [2] J.-Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker, description of the algorithm, 2000. Intel Corporation, Microprocessor Research Lab, http://robots.stanford.edu/cs223b04/algo_tracking.pdf.
- [3] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc 12th IEEE International Symposium on Wearable Computers*, pages 15–22, 2008.
- [4] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the IEEE conference Computer Vision and Pattern Recognition*, pages 142–149. IEEE Computer Society, Los Alamitos, USA, 2000.
- [5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 681–685, 2001.
- [6] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *IJCV*, 10(2):101–124, 1993.
- [7] T. Dickscheid, F. Schindler, and W. Förstner. Coding images with local features. *International Journal of Computer Vision*, pages 1–21, 2010. published on-line, April 2010.
- [8] M. Donoser, H. Bischof, and M. Wiltsche. Color blob segmentation by MSER analysis. In *Proceedings of the International Conference on Image Processing*, pages 757–760, Atlanta, USA, 2006. IEEE, Los Alamitos, USA.
- [9] F. Fraundorfer and H. Bischof. Evaluation of local detectors on non-planar scenes. In *Proc. of 28th Workshop of the Austrian Association for Pattern Recognition (ÖAGM/AAPR)*, pages 125–132, Österreichische Computer Gesellschaft 3-85403-179-3, Hagenberg, 2004.
- [10] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of 4th Alvey Vision Conference*, pages 147–151, March 1988.
- [11] F. Jurie and M. Dhome. Real time robust template matching. In *Proceedings of the British Machine Vision Conference (BMVC2002)*, 2002.
- [12] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the 8th European Conference on Computer Vision*, volume 1 of *LNCS 3021*, pages 228–241, Prague, May 2004. Springer-Verlag.

- [13] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.
- [14] Y. Li, H. Ai, T. Yamashita, S. Lao, , and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, USA, 2007.
- [15] M. Loog and F. Lauze. The improbability of Harris interest points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(06):1141–1147, June 2010.
- [16] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of IEEE International Conference on Computer Vision (ICCV1999)*, pages 1150–1157. IEEE Computer Society, 1999.
- [17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [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 (IJCAI '81)*, pages 674–679, 1981.
- [19] J. Matas, O. Chum, U. M., and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In P. L. Rosin and D. Marshall, editors, *Proc. of the British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002. BMVA.
- [20] J. Maver. Self-similarity and points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(07):1211–1226, July 2010.
- [21] K. Mikolajczyk and C. Schmid. Scale & affine invariant point detector. *International Journal of Computer Vision*, 60(1):63–86, 2004.
- [22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(7):43 – 72, November 2005.
- [23] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. of The 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, Massachusetts*, page 584. IJCAI, August 1977.

- [24] D. Nistér, O. Naroditsky, and J. R. Bergen. Visual odometry. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pages 652–659. IEEE Computer Society, 2004.
- [25] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, USA, 2007.
- [26] T. Pajdla and V. Hlaváč. Zero phase representation of panoramic images for image based localization. In F. Solina and A. Leonardis, editors, *Proc. of 8-th International Conference on Computer Analysis of Images and Patterns*, number 1689 in Lecture Notes in Computer Science, pages 550–557, Tržaška 25, Ljubljana, Slovenia, September 1999. Springer Verlag.
- [27] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511, October 2005.
- [28] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.
- [29] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or ‘how do i organize my holiday snaps?’. In *Proceedings of the 7th European Conference on Computer Vision*, volume 1 of *LNCS 2350*, pages 414–431. Springer-Verlag, 2002.
- [30] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
- [31] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE Computer Society, Los Alamitos, USA, 1994.
- [32] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, Carnegie Mellon University, April 1991.
- [33] T. Tuytelaars and L. V. Gool. Content-based image retrieval based on local affinity invariant regions. In *International Conference on Visual Information Systems*, pages 493–500, 1999.
- [34] T. Tuytelaars and L. V. Gool. Matching widely separated views based on affine invariant regions. *International Journal on Computer Vision*, 59(1):61–85, 2004.

- [35] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.
- [36] www.willowgarage.com. OpenCV (open source computer vision) library.
- [37] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 677–692, 2009.
- [38] O. Zuniga and R. Haralick. Corner detection using the facet model. In *Computer Vision and Pattern Recognition*, pages 30–37, Los Alamitos, CA, 1983. IEEE.

A Resumé

Disertační práce navrhuje detektor skvrn v intenzitním obraze postupem, který je nazván detektor stabilní vlnou (Stable Wave detector, dále SWD). Skvrnami jsou přibližně eliptické oblasti, které odpovídají lokálním extrémům jasové funkce, tedy jsou světlejší nebo tmavší než jejich okolí. Skvrny nemají žádnou sémantiku. Jejich hlavní předností je, že se v obraze daří nalézat spolehlivě a opakovaně. V tom se skvrny podobají významným bodům, např. Harrisovým rohům. Předností SWD je robustnost, přesnost lokalizace a rychlost.

K detekci a lokalizaci skvrn se používá fáze první harmonické v lokálním obrazem posouvaném okně. Lokalizace je výsledkem integračního postupu, což SWD odlišuje od většiny jiných detektorů, které se opírají o derivování. To přináší robustnost, odolnost vůči šumu a rozmazání v obraze. SWD lokalizuje skvrnu s podpixlovou přesností, poskytuje odhad přesnosti spolu s odhadem síly a velikosti skvrny. Mezi skvrnami je přirozeně zavedena minimální vzdálenost, která je daná periodou vlny používanou při detekci. Této vzdálenosti lze využít pro zjednodušení úlohy párování mezi více skvrnami ve scéně. SWD je spolehlivý nástroj, ale sám osobě není v detekci a lokalizaci lepší než ostatní špičkové metody.

Situace se ovšem dramaticky zlepší, když se myšlenka SWD detektoru použije při sledování (angl. tracking) cílů ve videosekvenci. Zde se stanovuje korepondence mezi body, v našem případě skvrnami, napříč jednotlivými snímky videosekvance. Výstupem je ultrarychlý sledovač (tracker), který předčí nejlepší známé metody v této třídě úloh. Sledovač pracuje se dvěma vstupy: (a) polohou skvrny v předchozím snímku, která slouží jako odhad polohy skvrny v současném snímku a (b) novým snímekem, v němž se skvrna hledá. Úloha sledování je vyjádřena v několika úrovních abstrakce. Na nižší úrovni se korepondence hledají nezávisle na sobě napříč různými měřítky. Sledování na vyšší úrovni abstrakce bere v úvahu prostorové uspořádání skvrn a predikuje jejich novou polohu. Hledá se kompromis mezi přesností a rozsahem povolených posunů skvrn, a to v různých měřítcích.

SWD detektor a na něm založený sledovač ve videosekvencích prošel rozsáhlým experimentálním ověřením. Implementace algoritmů jazyce C je poskytnuta formou otevřeného software výzkumné komunitě.

B Author's publications

Conference papers relevant to thesis:

- [P1] Jan Dupač and Václav Hlaváč. Stable Wave Detector of Blobs in Images. In *Proceedings of the 28th DAGM (German Pattern Recognition Society) Symposium*, Berlin, Germany, September 2006. Springer, Lecture Notes in Computer Science, pages 760-769
- [P2] Dupač, Jan and Matas, Jiří Ultra-fast Tracking Based on Zero-shift Points. *Proceedings of the 36th International Conference on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, May 2011, accepted, IEEE Signal Processing Society.

Technical Reports relevant to thesis:

- [P3] Dupač, Jan Phd Thesis Proposal: 3D computer vision-based localization Research Report CTU–CMP–2005–02. Center for Machine Perception, K13133 FEE Czech Technical University in Prague, Czech Republic, January 2005.
- [P4] Dupač, Jan and Hlaváč, Václav. Stable Wave Detector for Precise and Fast Detection of Blobs in The Image. Research Report CTU–CMP–2006–03. Center for Machine Perception, K13133 FEE Czech Technical University in Prague, Czech Republic, April 2006.

Others:

- [P1] Dupač, J. and Bláha, and J. Zástěra, and M. Horák, Z. Detekce a analýza stopových množství explozivních materiálu a přenosný detektor/analyzátor k provádění detekce a analýzy. Patent č: 298856, Úřad průmyslového vlastnictví, Praha, patent udělen 17.1.2008
- [P1] J. Dupač and C. Uematsu and M. Shen, V Hlaváč and H. Kambara, K. Okano Precise Gene Expression Measurement with Outlier Detection. In *Proceedings of The 13th International Conference on Genome Informatics GIW2002*, Tokyo, Japan, December 2002, Japanese Society for Bioinformatics.
- [P1] Dupač, Jan and Hlaváč, Václav. GEPCLUST Tool for Clustering Gene Expression Profiles. Research Report CTU–CMP–2001–17. Center for Machine Perception, K13133 FEE Czech Technical University in Prague, Czech Republic, May 2001.

