

# Uncertainty Model for Template Feature Matching

Hongmou Zhang\*, Denis Griebach, Jürgen Wohlfeil, Anko Börner

German Aerospace Center, Rutherfordstrae.2, 12489 Berlin

**Abstract.** Using visual odometry and inertial measurements, indoor and outdoor positioning systems can perform an accurate self-localization in unknown, unstructured environments where absolute positioning systems (e.g. GNSS) are unavailable. However, the achievable accuracy is highly affected by the residuals of calibration, the quality of the noise model, etc. Only if these unavoidable uncertainties of sensors and data processing can be taken into account and be handled via error propagation, which allows to propagate them through the entire system. The central filter (e.g. Kalman filter) of the system can then make use of the enhanced statistical model and use the propagated errors to calculate the optimal result. In this paper, we focus on the uncertainty calculation of the elementary part of the optical navigation, the template feature matcher. First of all, we propose a method to model the image noise. Then we use Taylor's theorem to extend two very popular and efficient template feature matchers sum-of-absolute-differences (SAD) and normalized-cross-correlation (NCC) to get sub-pixel matching results. Based on the proposed noise model and the extended matcher, we propagate the image noise to the uncertainties of sub-pixel matching results. Although the SAD and NCC are used, the image noise model can be easily combined with other feature matchers. We evaluate our method by an Integrated Positioning System (IPS) which is developed by German Aerospace Center. The experimental results show that our method can improve the quality of the measured trajectory. Moreover, it increases the robustness of the system.

**Keywords:** Uncertainty model, Image noise model, Template matching, Propagation of uncertainty, Sub-pixel matching

## 1 Introduction

Uncertainty occurs in almost every element of a computer vision system. For example, the measurements of sensors contain noise, their calibration is affected with errors, etc. In order to archive high accuracy, the uncertainty of each element should be taken into account in computer vision systems. Only if the uncertainties are propagated correctly through the whole system, the central

---

\* Corresponding author

filter (e.g. Kalman filter) of the system can make use of them to calculate the optimal result by a statistical model.

In this paper, we focus on the uncertainty of the image (i.e. image noise) and the uncertainty of the sub-pixel template feature matching. We propose an image noise model which can be used for real-time processing. On the other hand, the noise model can be combined with our proposed sub-pixel matching algorithm to calculate the uncertainty of template matching without a significant computational overhead. Particular uncertainty calculation methods are presented for the SAD and NCC matcher. However, it can easily be ported to other template feature matching algorithms.

Feature extraction and feature matching are elementary parts in many computer vision applications, such as optical navigation system (e.g. SLAM [20, 24, 4], IPS [11, 10]). In these applications, features are extracted from one image by a feature extractor (e.g. FAST, AGAST [26, 23, 32]). A feature matcher (e.g. SAD, NCC, KLT [12, 28, 21]) matches the features to another image. The image noise affects the matching result and therefore influences the performance of the whole system. Many researches focus on the problem of feature uncertainties [31, 16, 15, 27].

In [31] the authors present a framework to calculate the uncertainty of scale invariant features (SIFT) [19] and speeded up robust features (SURF) [3]. In that paper, the uncertainty of features are depend on the scale and the neighborhood. The results of the experiments show that the proposed method improves the performance for bundle adjustment. However, due to the complexity of calculating SIFT features, this method possibly may not be used for real-time computer applications, especially, for mobile platforms.

In [16] the authors proposed a method for involving the uncertainty of features into a homography and fundamental matrix calculation. The paper shows that in most of the cases the results can be improved by considering the covariances of the features. However, because of lacking a noise model, the method can only get rough uncertainties of features.

In our method, we assume that the covariance for the feature extraction step is zero, because the template feature matcher gets pixel by pixel matching results. Therefore the uncertainties of feature extraction can be omitted and the noise in both images impacts the feature matching step. By combined with our proposed image noise model, the template feature matcher provides matching results and covariance matrices with values propagated from image noise. This approach simplifies the calculation and enables real-time processing without losing accuracy. As shown in section 4, the uncertainty can be used to identify and eliminate features with high uncertainties, usually indicating mismatched features. The most important benefit is that the uncertainty of the matching can be involved in further calculations, e.g. triangulation, ego-motion calculation of the system, etc. The propagation of the uncertainties through the whole calculation chain to the central filter significantly increases the stability and the quality of the results.

This paper is organized as follows: In section 2 an introduction to our image noise model is given. Such a model can be used to calculate the uncertainties of sub-pixel matching results as well as for further processing. In section 3 a sub-pixel template matching algorithm and a method to propagate uncertainties from image noise to sub-pixel matching results are described. Experimental results are presented in section 4, and section 5 concludes the paper.

## 2 Image noise model

Even though it varies between cameras and scenes, image noise is always present in images taken with digital cameras. There are two major sources of noise. Firstly, fixed pattern noise is caused by different light sensitivities (photo response non-uniformity - PRNU [2]) and signal offsets (dark signal non-uniformity - DSNU [8]) of the pixels of an image sensor. This noise does not change over short time and is usually corrected by the camera itself. Secondly, dynamic noise changes from image to image even without a change of the input signal. It is mainly caused by the read-out electronics (read-out noise [25]), but also by the stochastic nature of the incoming photons (photon noise). This is just a simple model, more accurate one's can be applied if needed.

To control the impact of image noise to image processing, it can be taken into account as the uncertainty of an image. Such uncertainties can be handled via propagation of uncertainties, which allows to propagate them through the entire computer vision system. In order to achieve this goal, the mathematical model of the image noise must be known. There are many noise models proposed in computer vision community, e.g. [5]. However, they either do not quantify the noise (e.g. Salt and Pepper Noise [14]) or are built as common probability models (e.g. Gaussian Noise [9]), which do not take into account the variation of different camera systems.

In this paper, we propose a method to build an image noise model which is suitable for real-time processing. We assume that the parameters of that noise model are different for each camera so that the model building step can be done by camera calibration step. During the calibration, by taking a batch of  $M$  frames (e.g. 100) in a short time (less than 1 minute) from a fixed scene,  $M$  almost identical images are received. The mean value and standard deviation of all pixels are calculated for all values measured during the  $M$  frames. Hence:

$$\text{Mean}(i, j) = \frac{1}{M} \sum_{m=1}^M g(i, j)_m \quad (1)$$

where  $g(i, j)_m$  is the locale gray value at coordinate  $(i, j)$  for the frame  $m$ . The standard deviation of this input vector can be easily calculated. The relation between all mean values and standard deviations can be displayed in a graph, which is done in fig. 1 for a set of sample images. The standard deviations can be seen as a function of the corresponding mean values. It is obvious that the standard deviation grows according to the mean gray value. This reveals the

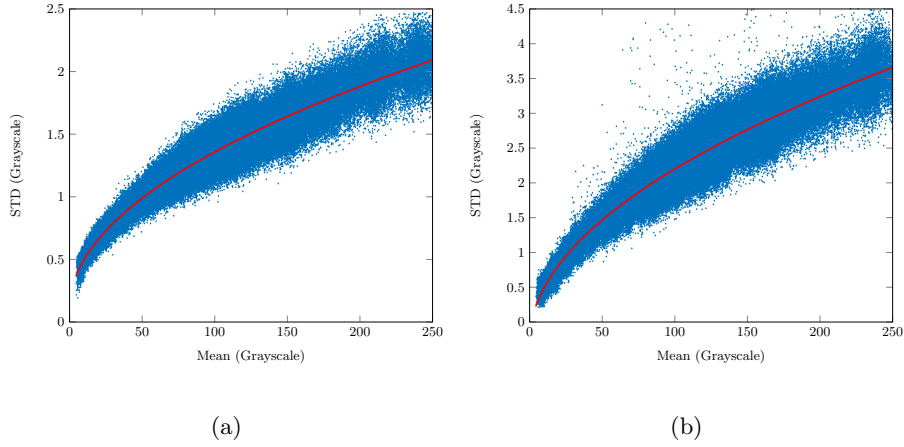


Fig. 1: In the left: pixels mean and the corresponding standard deviation are shown by blue points. The red curve is the fitted model based on our proposed noise equation, with  $G = 58.1224$  and  $N_E = 0.3166$ . In the right: by comparison with left side, the input signal of camera is amplified by 10dB, from decibel equation  $G_{dB} = 20 \log_{10}(V/V_0)$ , the ratio between  $G$  in (a) and (b) is 3.162. The fitted new noise model is  $G = 18.1069$  and  $N_E = 0.6453$ , the ratio between two  $G$  is 3.21, it is very close to the theory value. The camera is Prosilica GC1380H.

relation between pixel noise and the gray value of the pixel. On the other hand, because of electronic noise, the standard deviation should be greater than zero even for dark pixels. Based on this, we propose a noise model for each pixel.

$$\text{Noise} = \sqrt{N_E^2 + I/G} \quad (2)$$

$N_E^2$  is the variance of electronic noise of camera in gray scale values, the second part is from shot noise (Shot noise =  $\sqrt{\text{signal}}$  [13]), here  $I$  is the gray value of the pixel and  $G$  is a gain parameter. Next, a Gauss-Newton algorithm is used to estimate the model with the mean values and standard deviations from the calibration images, fitting the curve (2) to the data. Eq. (3) shows the Gauss-Newton algorithm.

$$\beta^{(s+1)} = \beta^{(s)} - (J_r^T J_r)^{-1} J_r^T r(\beta^{(s)}) \quad (3)$$

The Gauss-Newton algorithm iteratively calculates the results. In equation 3,  $\beta$  is the vector of variables to calculate (in our case  $N_E$  and  $G$ ), the superscript  $s$  indicate the  $s$  th iteration value.  $J_r$  is the Jacobian matrix of a residual function  $r(\beta)$ , where

$$\begin{aligned} r(\beta) &= y - f(I, \beta) \\ J_r &= \frac{\partial r(\beta)}{\partial \beta} \end{aligned} \quad (4)$$

In our case  $y$  is the calculated standard deviation of the gray values of the image set, and the  $f(I, \beta)$  is the proposed noise model (eq. (2)). Starting with an initial  $\beta^{(0)}$ , the Gauss-Newton algorithm can get a convergent  $\beta$  (fig. 1) after several iterations. More details can be found at [22]. Knowing the parameters  $G$  and  $N_E$  makes the noise model available. This noise information can be used in the further processing, e.g. to model the uncertainty of template matching introduced in section 3.2.

### 3 Error model for template sub-pixel matching algorithm

Once the image noise model is known, it can be involved into data processing chain to get the uncertainty information. In this section, we focus on the uncertainty of feature matching result based on proposed noise model. The calculated uncertainty of matching result can be easily propagated to further calculation.

The SAD and NCC template feature matchers [1, 6] are extensively used in stereo feature matching, feature tracking, etc. Therefore in this paper, these two feature matchers are used. Moreover, we extend them by Taylor’s theorem to get sub-pixel level matching results. Sub-pixel matching algorithm is a well-researched topic in computer vision community, many sub-pixel matching algorithms are proposed [21, 30, 17]. Besides, descriptor based feature extraction algorithm [19, 3, 18] can get sub-pixel matching result as well. All of these methods can get good sub-pixel matching results. However, these methods need iteration calculation, image pyramid, difference of Gaussian, brute-force search respectively, which are too “heavy” for low computational resource platform. Our method is derived from polynomial interpolation which is faster and easy to implement and can get acceptable results, on the other hand, the uncertainty of matching result can be calculated by combination with our noise model.

#### 3.1 Template sub-pixel matching algorithm

The sub-pixel matching algorithm includes three steps. At first, the common SAD/NCC template feature matcher is used to match the features, the details are shown below. For each extracted feature on the first image, a  $5 \times 5$  pixel region around it is taken as a template. Then the template is shifted over the second image pixel by pixel. For each position, the SAD/NCC value between the template and covered area in the second image is calculated. The search area in the second image is limited by some conditions (e.g. epipolar line). The position in the second image with the lowest SAD value or the highest NCC value is the matched feature. This step outputs the matching results with pixel-level coordinates.

Secondly, in case that the search area in the first step does not cover all the  $3 \times 3$  pixel neighbor positions around the matched feature (e.g. epipolar line case), the SAD/NCC values for these positions need to be calculated by the same template. Then a  $3 \times 3$  SAD/NCC matrix  $\mathbf{A}$  can be picked up, as shown in fig. 2.

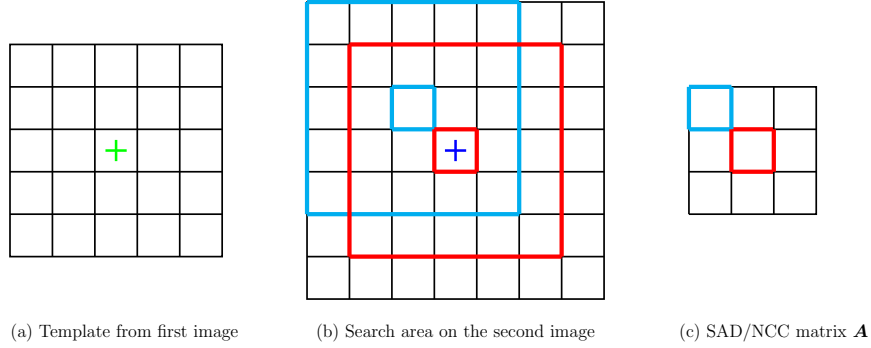


Fig. 2: In (a), a  $5 \times 5$  template is taken from first image, the green cross represents the feature position. In (b), part of search area on the second image is shown, the blue cross indicates the matched feature position, the SAD/NCC value between the template and the covered area (red part) is stored in the central entry of matrix  $\mathbf{A}$  in figure (c). By sliding the template center around the  $3 \times 3$  neighbor area of the matched feature in (b) (as indicated by different colors), all the values in the matrix  $\mathbf{A}$  can be calculated. The template and all the  $7 \times 7$  pixels in figure (b) are needed for the propagation of uncertainties step which is described in section 3.2.

In the third step, the values of matrix  $\mathbf{A}$  can be seen as a surface in 3D space with the central element located in a valley (SAD) or on a hill (NCC). Using a two-dimensional interpolation of the values allows to get a more precise position of the surfaces extremum: the sub-pixel position of the matched feature. Two-dimensional interpolation follows the second-order Taylor series in section 3.1.

$$f(x + \delta x, y + \delta y) \approx f(x, y) + \left[ \frac{\partial f}{\partial x} \delta x + \frac{\partial f}{\partial y} \delta y \right] + \frac{1}{2} \left[ \frac{\partial^2 f}{\partial x^2} (\delta x)^2 + 2 \frac{\partial^2 f}{\partial x \partial y} \delta x \delta y + \frac{\partial^2 f}{\partial y^2} (\delta y)^2 \right] \quad (5)$$

In section 3.1,  $f(x, y)$  is the matrix  $\mathbf{A}$ , where  $x$  and  $y$  are the feature coordinates. If the feature coordinate is expressed in vector form  $\mathbf{x} = [x \ y]^\top$ , then the equation can be written as:

$$f(\mathbf{x} + \delta \mathbf{x}) \approx f(\mathbf{x}) + \left( \frac{df}{d\mathbf{x}} \right)^\top \delta \mathbf{x} + \frac{1}{2} \delta \mathbf{x}^\top \frac{d^2 f}{d\mathbf{x}^2} \delta \mathbf{x} \quad (6)$$

The local extremum is calculated by eq. (7)

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = 0 \quad (7)$$

this leads to:

$$\delta \hat{\mathbf{x}} = - \left( \frac{d^2 f}{d\mathbf{x}^2} \right)^{-1} \frac{df}{d\mathbf{x}} \quad (8)$$

Equation (8) outputs two real numbers, taken as an offset for the feature coordinate  $x$  and  $y$ . Adding these two values to the feature pixel level coordinate will get the sub-pixel coordinate of the feature.

### 3.2 Propagation of uncertainty of the SAD feature matcher

In this subsection, the proposed image noise model is applied to the SAD sub-pixel matching algorithm. The aim is to have an uncertainty model of the matching procedure. The equation of SAD is shown as follows.

$$\text{SAD}(u, v) = \sum_{i,j} |\mathcal{S}(u+i, v+j) - \mathbf{T}(i, j)| \quad (9)$$

$\mathcal{S}(u+i, v+j)$  is the search area in the second image,  $\mathbf{T}$  is the template from the first image. For each matched feature pair, the SAD propagation of uncertainties algorithm includes two parts. The first part propagates the image noise to the uncertainty of the  $3 \times 3$  matrix  $\mathbf{A}$ . The second part handles the uncertainty of  $\mathbf{A}$  and finally gets the uncertainty of the sub-pixel matching result.

Part 1: From the linear propagation of uncertainties theory [7] it is known that in order to propagate the uncertainty; a matrix  $\mathbf{F}$  which can linearize the SAD calculation must be known.

$$\mathbf{a} = \mathbf{F}\mathbf{v} \quad (10)$$

Referring to fig. 2, the  $9 \times 1$  vector  $\mathbf{a}$  of eq. (10) is the reformatted  $3 \times 3$  matrix  $\mathbf{A}$ . The vector  $\mathbf{v}$  includes all the pixel values from the template and the  $7 \times 7$  search area on the second image. The method for specifying  $\mathbf{F}$  is described in the following.

The template from the first image is reformatted, from a  $5 \times 5$  matrix  $\mathbf{V}_f$  into a  $25 \times 1$  vector  $\mathbf{v}_f$ . The corresponding  $7 \times 7$  area  $\mathbf{V}_s$  (see fig. 2) in the second image is reformatted into a  $49 \times 1$  vector  $\mathbf{v}_s$ .

By concatenation of these two vectors, a  $74 \times 1$  vector  $\mathbf{v} = [\mathbf{v}_s \ \mathbf{v}_f]^T$  (see fig. 3) is defined and used for further calculations. Furthermore, a  $9 \times 74$  matrix  $\mathbf{F}$  (see fig. 3) is built. In the first,  $\mathbf{F}$  is set to be a zero matrix (i.e. all of the entries equals zero). Then, some entries of  $\mathbf{F}$  are calculated as follows:

$$\begin{aligned} & \mathbf{F}_s[(i+m-1+7(j+n-2)), (m+3(n-1))] \\ & \quad = \text{sgn}[\mathbf{V}_s(i+m-1, j+n-1), \mathbf{V}_f(i, j)] \\ & \mathbf{F}_f[(i+5(j-1)), (m+3(n-1))] \\ & \quad = -\text{sgn}[\mathbf{V}_s(i+m-1, j+n-1), \mathbf{V}_f(i, j)] \end{aligned} \quad (11)$$

with  $1 \leq i, j \leq 5, \quad 1 \leq m, n \leq 3$

where  $\mathbf{V}_f(i, j)$  indicates the entry which is located in the  $i^{\text{th}}$  column and  $j^{\text{th}}$  row in matrix  $\mathbf{V}_f$ .

The  $9 \times 49$  matrix  $\mathbf{F}_s$  is the left part of  $\mathbf{F}$  and the  $9 \times 25$  matrix  $\mathbf{F}_f$  is the right part, as shown in fig. 3. Parameters  $m$  and  $n$  are shift coordinates of

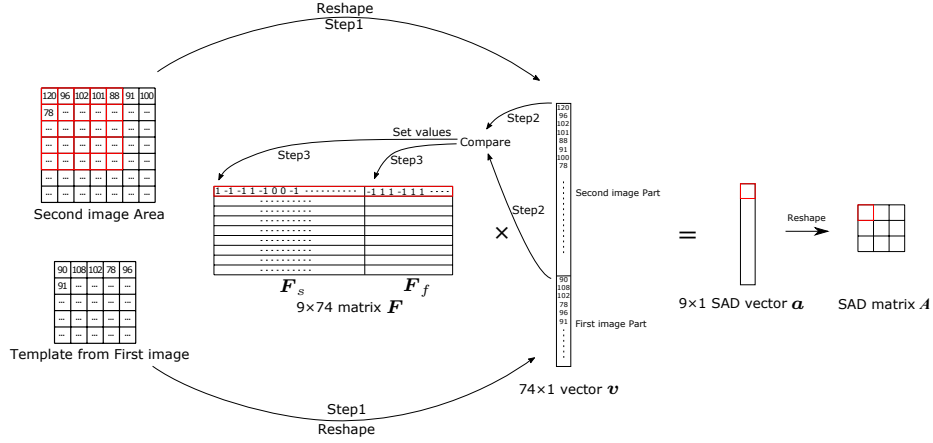


Fig. 3: First-image and second-image areas are reformatted first, forming a  $74 \times 1$  vector  $\mathbf{v}$ . Next, values in  $\mathbf{F}$  are calculated by Eq. (12). Finally, SAD vector  $\mathbf{a}$  equals  $\mathbf{F}\mathbf{v}$ . The  $3 \times 3$  matrix  $\mathbf{A}$  is obtained by reformatting  $\mathbf{a}$ ; this matrix is identical to the results of the original SAD algorithm. The red part of the second-image area shows the area considered for the calculation of the first SAD value. The red part in  $\mathbf{F}$  reflects the relationship between the first-image area and the red part of the second-image area.

the template  $\mathbf{V}_f$  over the search area  $\mathbf{V}_s$  in horizontal direction and vertical direction, respectively.

For example, the blue area in fig. 2(b) indicates the shift coordinate (1, 1), and the red area indicates (2, 2). The sign function  $\text{sgn}$  is defined as follows:

$$\text{sgn}(x, y) := \begin{cases} -1 & \text{if } x < y \\ 1 & \text{if } x > y \end{cases} \quad (12)$$

The template  $\mathbf{V}_f$  can only cover a part of search area  $\mathbf{V}_s$ , for each  $(m, n)$ ; some entries in  $\mathbf{F}_s$  remain unchanged as 0. Therefore, the values in  $\mathbf{F}$  are from the set  $\{-1, 0, +1\}$ . Finally, we obtain a dynamically changing matrix  $\mathbf{F}$ , as the values of  $\mathbf{F}$  are different for different features. These steps guarantee that the vector  $\mathbf{a}$  in eq. (10) is always identical to the standard absolute calculation results.

Once we know the matrix  $\mathbf{F}$ , the SAD algorithm becomes a linear calculation. From our proposed noise model, the noise vector  $\mathbf{v}_n$  of all pixels in  $\mathbf{v}$  can be calculated. Assuming, that the noise level between each pixel is uncorrelated, a  $74 \times 74$  covariance matrix  $\Sigma_{\mathbf{v}}$  is defined. Its diagonal elements are  $\mathbf{v}_n^2$ , and the others are 0. The covariance of  $\mathbf{a}$  can be calculated as:

$$\Sigma_{\mathbf{a}} = \mathbf{F}\Sigma_{\mathbf{v}}\mathbf{F}^{\top} \quad (13)$$

Hence,  $\Sigma_{\mathbf{a}}$  is the  $9 \times 9$  covariance matrix of  $\mathbf{a}$ .



Part 2: the task of the second part is the propagation of uncertainties from  $\Sigma_{\mathbf{a}}$  to the final sub-pixel calculation results. From eq. (6) and eq. (8), the sub-pixel calculation step is a non-linear function. The Jacobian matrix of eq. (8) is calculated. First of all, eq. (8) can be written as:

$$\delta\hat{\mathbf{x}} = \begin{bmatrix} -(\partial yy\partial x - \partial xy\partial y)/(\partial xx\partial yy - \partial xy\partial xy) \\ -(\partial xx\partial y - \partial xy\partial x)/(\partial xx\partial yy - \partial xy\partial xy) \end{bmatrix} \quad (14)$$

where

$$\begin{aligned} \partial x &= \frac{a_{23} - a_{21}}{2} & \partial xx &= a_{23} + a_{21} - 2a_{22} \\ \partial y &= \frac{a_{32} - a_{12}}{2} & \partial yy &= a_{32} + a_{12} - 2a_{22} \\ \partial xy &= \frac{a_{33} - a_{31} - a_{13} + a_{11}}{4} \end{aligned} \quad (15)$$

$a_{ij}$  is the element of matrix  $\mathbf{A}$  located in row  $i$  and column  $j$ . The Jacobian matrix is calculated as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial\hat{\mathbf{x}}}{\partial a_{11}} & \frac{\partial\hat{\mathbf{x}}}{\partial a_{12}} & \cdots & \frac{\partial\hat{\mathbf{x}}}{\partial a_{33}} \end{bmatrix} \quad (16)$$

$\mathbf{J}$  is a  $2 \times 9$  matrix, so the last step of calculation is:

$$\Sigma_{\delta\hat{\mathbf{x}}} = \mathbf{J}\Sigma_{\mathbf{a}}\mathbf{J}^T \quad (17)$$

$\Sigma_{\delta\hat{\mathbf{x}}}$  is a  $2 \times 2$  covariance matrix, the diagonal elements are the variance values of the matched sub-pixel coordinate, and the off-diagonal entries are the covariance values of the sub-pixel coordinates. This matrix finally includes the uncertainty information propagated from the image noise to the SAD sub-pixel matching result and can be used for further processing.

### 3.3 Propagation of uncertainty of the NCC feature matcher

The normalized cross correlation (NCC) template matching algorithm is very similar to the SAD algorithm, NCC uses the eq. (18) to calculate the normalized cross correlation between the templates from both images.

$$\text{NCC}(u, v) = \frac{1}{n} \sum_{i,j} \frac{(\mathbf{S}(u+i, v+j) - \bar{\mathbf{S}})(\mathbf{T}(i, j) - \bar{\mathbf{T}})}{\sigma_{\mathbf{S}}\sigma_{\mathbf{T}}} \quad (18)$$

In the equation,  $\mathbf{S}(u+i, v+j)$  is the search area in the second image,  $\mathbf{T}$  is the template from the first image, where  $n = i \cdot j$  and  $\bar{\mathbf{S}}, \bar{\mathbf{T}}$  are the means of the template and the search area. The standard deviation of  $\mathbf{S}$  and  $\mathbf{T}$  are represented by the symbols  $\sigma_{\mathbf{S}}, \sigma_{\mathbf{T}}$ . Because of the non-linear nature of NCC, the error propagation for the  $3 \times 3$  NCC matrix  $\mathbf{A}$  needs to be done by a Jacobian matrix. Considering an example, the template size is the same as in fig. 3. Hence,

the Jacobian matrix is calculated as follows:

$$\mathbf{J}_{ncc} = \frac{\partial f}{\partial p} = \begin{bmatrix} \frac{\partial f_1}{\partial p_{r1}} & \frac{\partial f_1}{\partial p_{r2}} & \cdots & \frac{\partial f_1}{\partial p_{l25}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_9}{\partial p_{r1}} & \frac{\partial f_9}{\partial p_{r2}} & \cdots & \frac{\partial f_9}{\partial p_{l25}} \end{bmatrix} \quad (19)$$

$f$  is the common NCC equation given in eq. (18),  $p_{ln}$  are the pixel values from the first image template and  $p_{rn}$  are the pixel values in the  $7 \times 7$  search area in the second image.  $f_1$  maps the template and the search area to the first entry of the  $3 \times 3$  NCC matrix and so on. The usage of the  $9 \times 74$  matrix  $\mathbf{J}_{ncc}$  is same to the usage of  $\mathbf{F}$  in section 3.2. After the calculation of the covariance matrix  $\Sigma_a$ , the remaining steps are identical with part 2 in section 3.2. To avoid numerical errors, we recommend to use the Matlab symbolic calculation to get the final format of  $\mathbf{J}_{ncc}$ .

## 4 Experiment results

To check the quality of the proposed method, we designed three experiments. Without loss of generality, the first experiment quantitatively verifies the proposed algorithm. The second one is a general feature matching test, and the third experiment checks the proposed method on an optical navigation system IPS [11], which was developed at the German Aerospace Center (DLR).

The first test is designed to verify the uncertainties propagation. The testing method is similar to a Monte Carlo test. The difference is that we do not generate artificial noise and add to image. Instead, we take a set of images, these images include noise from the camera system. The details of the test are described below.

The first step is similar to the noise model calculation step: The stereo camera system takes 100 image pairs from a fixed scene in a short time. The image contents are almost the same, but affected by noise from the camera system. Next, a feature extractor detects features from the first left image, then a sub-pixel template matcher matches the features to the first right image (without uncertainties propagation step). This step is repeated for all of the image pairs, but the feature extraction step is skipped. Instead, the coordinates of features in the first left image are used. As features are located in the same coordinate during 100 frames, there are 100 different stereo matching results, the standard deviation of the resulting sub-pixel offset is calculated and drawn as a curve in fig. 4. Now these empirical results can be used to compare them with the propagated uncertainty which is calculated in the next step.

In the second step, only the first image pair is needed. We apply the sub-pixel template matcher and involve the propagation of uncertainties step in the first image pair. This way we get the uncertainties of matched features which are propagated from image noise. The fig. 4 indicates that the noise model and propagation of uncertainties algorithm shows an accurate reflection of the real uncertainties of the matching results.

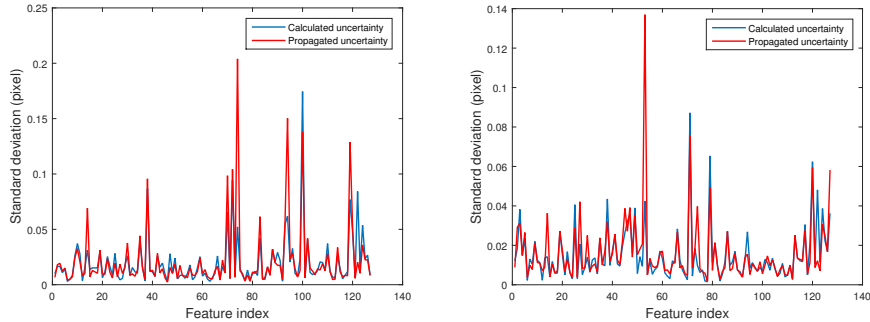


Fig. 4: The left graph shows the sub-pixel uncertainty of  $x$  coordinate, the blue curve is the uncertainty of features sub-pixel coordinate calculated from 100 images, and the red curve shows the propagated uncertainty based on our proposed method. The right graph shows the sub-pixel uncertainty of  $y$  coordinate. These two graphs show the strong correlation between real uncertainties and propagated uncertainties.

The second experiment check the performance of the proposed algorithm on stereo feature matching problem. First, the sub-pixel matching algorithm works on a stereo camera system. The noise model of the camera system is already calculated using our method. Fig. 5 shows the images from left and right camera respectively. We use the AGAST [23] feature extractor to get features from the left image, and a SAD template matcher to match the features to the right image under epipolar line constraint. The green and orange crosses on the left image are the features successfully matched to the right image by a common SAD template matcher, and the yellow and cyan crosses symbolize mismatches. However, the orange crosses are the features filtered out by our proposed sub-pixel matching algorithm because of their high uncertainty (higher than 0.4 pixel). And in fact the orange features in fig. 5 (near the cupboard and the white computer monitor) actually cannot be seen from the right camera's perspective. However, the common template matcher wrongly matches them to the right image (not drawn). Finally, the crosses on the left and right image are the features successfully matched after the sub-pixel matching step. This test shows that with the propagated uncertainty it is possible to filter for mismatched features, proving the correctness of the proposed algorithm from another perspective. The algorithm also improves the robustness of the system by filtering the mismatched features.

The last test based on an optical navigation project. The test platform is IPS. IPS was developed for real-time vision-aided inertial navigation [11, 10], especially for an environment where GNSS is not available. The IPS is a Kalman filter based optical navigation system, in the previous version, the normal NCC and SAD feature matcher are used for stereo matching and tracking respectively. The matching results are integral pixels. On the other hand, because lacking noise

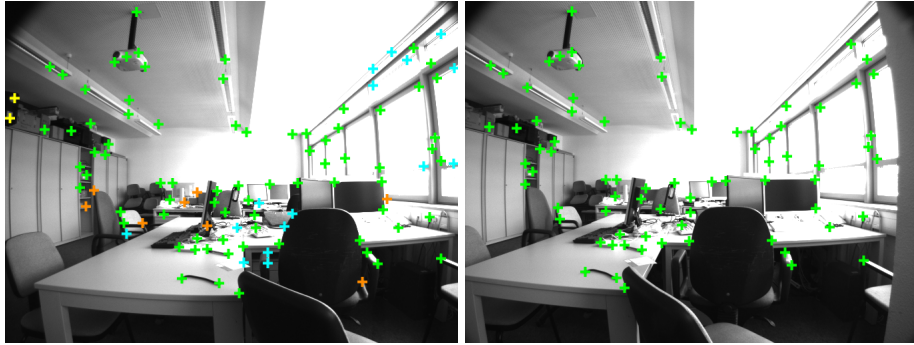


Fig. 5: Image pair from a stereo camera system, the crosses are the features. The orange and green crosses on the left image symbolize the successfully matched features by a normal template matcher, the yellow and cyan crosses are features where matching failed. The orange crosses are filtered out by the proposed SAD sub-pixel matching algorithm with error propagation.

model, we cannot get uncertainties of matching results. However in order to use Kalman filter, the uncertainties information must be provided, therefore in the previous version, only a rough uncertainty of matching result (e.g. quantization error ( $\frac{1}{12}$ ) [29]) are given. These problems can be solved by our proposed method. This experiment shows the comparison of measured trajectory in previous IPS version and the IPS combined with our noise model and sub-pixel algorithm.

For test purposes, a dataset is recorded by walking with the IPS through a realistic scene with a length of about 410 meters. Such a physical run is called a session. We recorded eight sessions in total. Because the lack of ground truth, the start and end position of the loop are exactly the same. As the system does only consider the motion information extracted from two consecutive image pairs, and it does not recognize that it has been in a place before, the performance of the system can be measured by the error between the known start position and the calculated end position. As a RANSAC algorithm is used for optical navigation, the calculated positions have a random component. This is why each session is processed (offline) 50 times to calculate the root-mean-square (RMS) of trajectory errors as a final result. More details about the test procedure can be found in [32].

The IPS gets state of the art results, the trajectory error is about 0.1% of the traveled distance. In this instance, usually it is hard to get improvement. However, as table (1) shows, the accuracy of the measurement is increased about 12% by our method. On the other hand, the new algorithm also leads to a better standard deviation, the improvement of standard deviation is about 44%, which means an improvement of the robustness of the system.

	Session:	1	2	3	4	5	6	7	8
Previous IPS	RMS(m)	0.68	0.25	0.62	0.29	0.52	0.49	0.39	0.44
	STD(m)	0.044	0.047	0.048	0.048	0.038	0.044	0.044	0.045
New IPS	RMS(m)	0.55	0.24	0.43	0.09	0.53	0.54	0.28	0.42
	STD(m)	0.016	0.023	0.026	0.018	0.016	0.019	0.017	0.023

Table 1: The 8 sessions trajectory errors. The RMS is the root-mean-square of errors of 50 runs, and STD is the standard deviation. The results show the sub-pixel matching algorithm decrease not only the trajectory errors but also the standard deviation. New IPS is the IPS combined with proposed methods.

## 5 Conclusion

In this paper, we propose a method to model the image noise, this noise model can be retrieved during normal camera calibration step. Based on the noise model, uncertainties propagation for sub-pixel matching algorithms is described. The proposed image noise model and the method to get uncertainty of sub-pixel matching results can be widely used in many computer vision applications. The performance of the proposed methods is evaluated by a full system test. The experimental results show that the noise model is actually able to reflect the uncertainty of sub-pixel matching results. An additional test shows that the uncertainty calculation can even be utilized as a mismatched filter without any computational overhead. The last test concentrates on the performance of the new algorithm combined with an optical navigation system. The result proves that the proposed method decreases the trajectory errors and standard deviation of errors simultaneously. The test shows our method can get significantly better results without much effort. In our future work, we will implement the uncertainties propagation method for other sub-pixel matching algorithms.

## References

1. Alsaade, F.: Fast and Accurate Template Matching Algorithm Based on Image Pyramid and Sum of Absolute Difference Similarity Measure. *Research Journal of Information Technology* 4(4), 204–211 (2012)
2. Amerini, I., Caldelli, R., Cappellini, V., Picchioni, F., Piva, A.: Estimate of prnu noise based on different noise models for source camera identification. *IJDCF*
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: *Computer vision ECCV 2006* (2006)
4. Bloesch, M., Omari, S., Hutter, M., Siegwart, R.: Robust visual inertial odometry using a direct EKF-based approach. In: *Intelligent Robots and Systems* (2015)
5. Boyat, A.K., Joshi, B.K.: A Review Paper: Noise Models in Digital Image Processing. *Signal & Image Processing: An International Journal* 6(2), 63–75 (2015)
6. Brunelli, R.: *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley (2009)
7. Clifford, A.: *Multivariate error analysis: a handbook of error propagation and calculation in many-parameter systems*. Wiley (1973)

8. Evtikhiev, N.N., Starikov, S.N., Cheryomkhin, P.A., Krasnov, V.V.: Measurement of noises and modulation transfer function of cameras used in optical-digital correlators. International Society for Optics and Photonics (2012)
9. Gonzalez, R.C., Woods, R.E.: Digital Image Processing (3rd Edition). Prentice-Hall, Inc. (2006)
10. Griebbach, D.: Stereo-Vision-Aided Inertial Navigation. Ph.D. thesis, Freie Universitt Berlin (2014)
11. Griebbach, D., Baumbach, D., Zuev, S.: Stereo-Vision-Aided Inertial Navigation for Unknown Indoor and Outdoor Environments. In: 2014 IPIN (2014)
12. Haralick, R., Shapiro, L.: Computer and Robot Vision. No. v. 2 in Computer and Robot Vision, Addison-Wesley Publishing Company (1993)
13. Holst, G.C.: CCD Arrays, Cameras, and Displays (2nd Edition) (1998)
14. Jayaraman: Digital Image Processing (2011)
15. Kanatani, K.i.: Uncertainty modeling and model selection for geometric inference. IEEE Transactions on Pattern Analysis and Machine Intelligence (2004)
16. Kanazawa, Y., Kanatani, K.: Do we really have to consider covariance matrices for image features? Electronics and Communications in Japan 86, 1–10 (2003)
17. Kim, K.B., Kim, J.S., Choi, J.S.: Fourier based image registration for sub-pixel using pyramid edge detection and line fitting. In: Intelligent Networks and Intelligent Systems, 2008. IEEE (2008)
18. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: ICCV. IEEE (2011)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60 (2004)
20. Lowry, S., Sunderhauf, N., Newman, P., Leonard, J.J., Cox, D., Corke, P., Milford, M.J.: Visual Place Recognition: A Survey. IEEE Transactions on Robotics (2016)
21. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2. IJCAI'81 (1981)
22. Madsen, K., Nielsen, H.B., Tingleff, O.: Methods for non-linear least squares problems (1999)
23. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: ECCV (2010)
24. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings IEEE ICRA (2007)
25. Nakamura, J.: Image Sensors and Signal Processing for Digital Still Cameras. Optical Science and Engineering, CRC Press (2016)
26. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. Pattern Analysis and Machine Intelligence (2010)
27. Sheorey, S., Keshavamurthy, S., Yu, H., Nguyen, H., Taylor, C.N.: Uncertainty estimation for KLT tracking. In: Asian Conference on Computer Vision (2014)
28. Shi, J., Tomasi, C.: Good features to track. In: CVPR (1994)
29. Stein, S., Jones, J.: Modern communication principles: with application to digital signaling. McGraw-Hill (1967)
30. Thevenaz, P., Ruttimann, U.E., Unser, M.: A pyramid approach to subpixel registration based on intensity. IEEE transactions on image processing (1998)
31. Zeisl, B., Georgel, P.F., Schweiger, F., Steinbach, E.G., Navab, N., Munich, G.: Estimation of Location Uncertainty for Scale Invariant Features Points. In: BMVC.
32. Zhang, H., Wohlfeil, J., Griebbach, D.: Extension and evaluation of the AGAST feature detector. In: XXIII ISPRS Congress annals 2016. vol. 3, pp. 133–137 (2016)