Covariance Estimation for SAD Block Matching

Johan Skoglund and Michael Felsberg

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden
{skoglund,mfe}@isy.liu.se

Abstract. The estimation of a patch position in an image is a long established but still relevant topic with many applications, e.g. pose estimation and tracking in image sequences. In most systems the position estimate needs to be fused with other estimates, and hence, covariance information is required to weight the different estimates in the right way. In this paper we address the issue with covariance estimation in the case of sum of absolute difference (SAD) block matching. First, we derive the theory for covariance estimation in the case of SAD matching. Second, we evaluate the suggested method in a virtual 3D patch tracking scenario in order to verify the performance in real-world scenarios.

1 Introduction

Motion information from images is useful for many applications. In this paper we focus on one type of method for doing this, called block matching. Typically a patch is given and the algorithm finds the best match in an image.

A common problem is merging of different measurements. Most common is to average a number of similar measurements to reduce average error but the measurements might also come from different sensors. Optimal merging of the measurements, requires knowledge about the accuracy for each measurement. Accuracy of these measurements and information whether the error in the measurements are dependent or not is stored in the "covariance matrix".

In this paper we present both the theory and also an evaluation of a method for estimating the covariance for block matching using *sum of absolute difference* (SAD), which is to our knowledge not yet an established topic.

1.1 Block Matching

Block matching is a common name for all algorithms that try to find the position of a patch p, in an image b. This can be done in a number of different ways but the goal is to find the minimum of an error function

$$\min_{\gamma} e(p, T(b, \gamma)) \; . \tag{1}$$

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 374–382, 2007.

The function e measures the difference between the patch and the image. A number of parameters γ, containing the patch position and possibly other interesting parameters like rotation of the patch or other shape information are estimated. To fully utilize the information both the pose γ and information about the accuracy, $\mathrm{cov}(\gamma)$ are needed. In this paper a dc-invariant error function based on the L_1 norm is used.

$$e_1 = \sum_{x,y} |(p(x,y) - \overline{p}) - (b(x,y) - \overline{b})| \tag{2}$$

This method, combined with fast subpixel interpolation is described in [1].

2 Covariance

When several measurements are combined, an assumption of the covariance of the different measurements is required. This assumption might be implicit, might assume that each measurement has the same covariance or explicit like Kalman filters. As an example we can look at a weighted linear least square problem:

$$\arg\min_x \|Ax = b\|_2 \tag{3}$$

Which solution is

$$x = (A^T w^{-1} A)^{-1} A^T w^{-1} b \tag{4}$$

Where w is the covariance matrix for b. Solving (3) can be classified into four groups depending on the assumption about the covariance matrix, from equal weight of each measurement to the full covariance:

1. Assume that all measurements are independent and with the same accuracy. w is the unit matrix.
2. Different accuracy for different measurements but that the measurements are independent. w is a diagonal matrix where each element that comes from one measurement is the same.
3. Assume that the measurements can be divided into groups where each measurement might influence all other in that group. Different groups are assumed to be independent. w is a block diagonal matrix with one block from each measurement.
4. One full covariance matrix for all measurements. This makes it possible to have different accuracy in different directions, and also that measurements are dependent. w is an arbitrarily positive definite matrix.

2.1 Covariance Derivation

Details of definition and computational laws for covariances can be found in many text books about statistics [2]. This section is therefore only a short summary containing the most important formulas needed for the rest of this paper. Practically, the covariance can be seen as a measurement of the spread of a stochastic vector x. The covariance contains both a measurement of the spread of different components in x and a measurement of the dependencies between the different parts. The covariance is defined as:

$$\mathrm{cov}[x] = E[(x - \overline{x})^T (x - \overline{x})] \tag{5}$$

The covariance matrix is a symmetric positive semidefinite matrix. Besides the definition, the rule for covariance propagation is needed. Covariance propagation is the calculation of the covariance of the output from a function based on the covariance of the input, estimating $cov[f(x)]$ from $cov[x]$. The exact solution of this problem can easily be found if f is a linear function, $f(x) = Ax + b$. In this case the covariance is:

$$\mathrm{cov}[f(x)] = A\mathrm{cov}[x]A^T \tag{6}$$

Finding the exact solution for an arbitrarily function is usually not possible and a common approximation, sometimes called the *Gauss approximation formula* [3], is to use a linear model of the function and approximate A with the Jacobian:

$$\mathrm{cov}[f(x)] \approx [\frac{d}{dx} f(E[x])] \mathrm{cov}[x] [\frac{d}{dx} f(E[x])]^T \tag{7}$$

3 Covariance Estimation

For each patch used in the tracking an estimate of the covariance is needed. The most basic form of confidence measure is the SAD distance between the patch and the image. However, this measurement has three important limitations:

- The measurement depends on intensity scaling in an unsuitable way, e.g. if the patch and image are scaled by 2 the error is scaled by 2.
- This gives an isotrop covariance. The certainty of the result might be different in different directions, i.e. we need a anisotropic measurement.
- SAD measures the distance between the patch and the image and gives a measurement of the similarity. Most applications do however need a measurement of the position accuracy, not the similarity.

Therefore, a more advanced covariance estimate is needed, an estimate which is able to model anisotrop covariances. This representation, makes it is also possible to represent the certainty for 1-D features like lines and edges. Figure 1 shows two examples of covariances estimated with the method proposed later. We can see that the accuracy is much higher perpendicular to the edge than along the edge.

The covariance can be estimated in two different ways [4]:

- Estimation from the structure of the error function around the minimum
- Estimation from the influence of each pixel in the patch

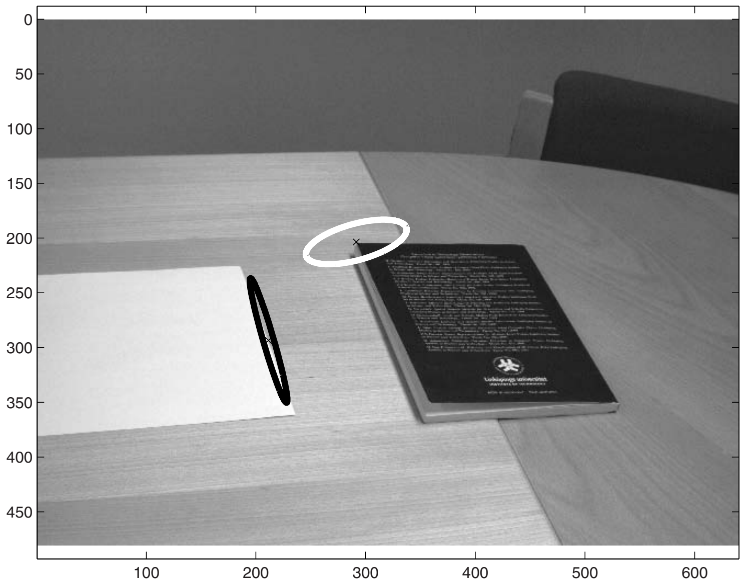

Fig. 1. Shape of uncertainties estimated for two patches

3.1 Covariance from Each Pixel

The most obvious way for estimating the covariance is probably to use covariance propagation (7). This requires knowledge about the covariance for each pixel and to differentiate the position of the minimum of the error wrt each pixel. One way to do this is explained in [5]. This paper shows that the covariance is estimated as:

$$\mathrm{cov}(\gamma) \approx [-\frac{de^2}{d\gamma^2}]^{-1}[\frac{de^2}{d\gamma dx}]\mathrm{cov}(x)[\frac{de^2}{d\gamma dx}]^T[-\frac{de^2}{d\gamma^2}]^{-T} \quad (8)$$

This method has been successfully used for error function like the L_2 norm [4]. It is however not possible to use this for the L_1 norm because the differentiation of the error function (2) wrt each pixel gives:

$$\frac{de}{dx} = sign(p - b) \; . \quad (9)$$

Differentiating (9) wrt γ gives 0 and that the whole covariance is 0. Therefore, the covariance needs to be estimated from the error function instead.

3.2 Covariance from the Error Function

The covariance can also be estimated from the structure of the error function around the minimum. We propose to apply (7) on the error function (2).

$$\mathrm{Var}[e(\gamma)] \approx [\frac{de}{d\gamma}]\mathrm{cov}(\gamma)[\frac{de}{d\gamma}]^T \; . \quad (10)$$

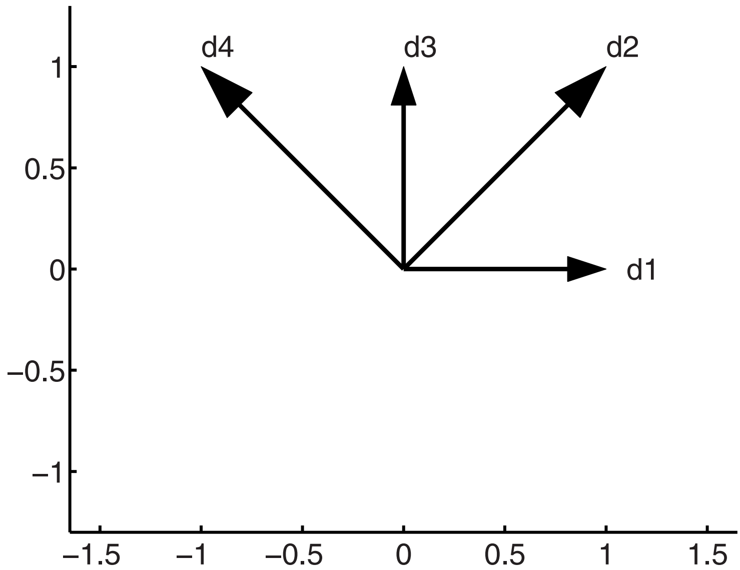

Fig. 2. Directions for the four derivatives d_1 ... d_4

Cov(γ) is a symmetric covariance matrix, hence a real-valued eigensystem decomposition exists and is given as

$$\mathrm{cov}[\gamma] = \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T \ . \tag{11}$$

Plugging this into (10) results in

$$\mathrm{Var}[e(\gamma)] \approx \ [\tfrac{df}{d\gamma}]^T(\lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T)[\tfrac{df}{d\gamma}] \tag{12}$$

$$= \lambda_1([\tfrac{df}{d\gamma}]^T e_1)^2 + \lambda_2([\tfrac{df}{d\gamma}]^T e_2)^2 \ . \tag{13}$$

Rewriting (13) using the Frobenius product $\langle\rangle_F$ [6]gives

$$\mathrm{Var}[e(\gamma)] \approx \lambda_1 \langle [\tfrac{df}{d\gamma}][\tfrac{df}{d\gamma}]^T | e_1 e_1^T \rangle_F + \lambda_2 \langle [\tfrac{df}{d\gamma}][\tfrac{df}{d\gamma}]^T | e_2 e_2^T \rangle_F \tag{14}$$

$$= \langle [\tfrac{df}{d\gamma}][\tfrac{df}{d\gamma}]^T | \mathrm{Cov}[\gamma] \rangle_F \ . \tag{15}$$

To be able to estimate the full covariance matrix, at least three different derivatives are needed [7]. Using derivatives in four directions is however useful since this makes it easy to sample the derivatives regularly. If the derivatives d_1 to d_4 are estimated in the x,y and the diagonal directions according to figure 2, these four responses correspond to the frame tensors

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{16}$$

$$B_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad B_4 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \ . \tag{17}$$

For the tensor computation we need the dual frame with minimum norm given as [7]

$$\tilde{B}_1 = \begin{bmatrix} 0.6 & 0 \\ 0 & -0.4 \end{bmatrix} \qquad \tilde{B}_2 = \begin{bmatrix} 0.2 & 0.25 \\ 0.25 & 0.2 \end{bmatrix} \tag{18}$$

$$\tilde{B}_3 = \begin{bmatrix} -0.4 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \tilde{B}_4 = \begin{bmatrix} 0.2 & -0.25 \\ -0.25 & 0.2 \end{bmatrix} . \tag{19}$$

Frame theory gives that the least square solution of (15) is [7]:

$$\mathrm{Cov}[\gamma] = \mathrm{Var}[e(\gamma)] \sum_{i=1}^{4} \tilde{B}_i |d_i|^{-2} . \tag{20}$$

To simplify the notation, we define the tensor

$$T = \sum_{i=1}^{4} \tilde{B}_i |d_i|^{-2} . \tag{21}$$

For the next step an assumption about the distribution of the errors $e(\gamma)$ is needed, how $Var[e(\gamma)]$ should be approximated from the minimum of the error function. In general, the estimate of the variance is:

$$\mathrm{Var}[e(\gamma)] = c\mathrm{E_{min}}^n . \tag{22}$$

where C and n depends on the assumed distribution. C and n are found by analyzing

$$\mathrm{Var}[e(\gamma)] = cE[e(\gamma)]^n \tag{23}$$

for the given distribution. Examples of c and n for different distributions can be found in table 1. Combining (20) and (22) gives that the covariance of γ can be estimated as:

$$\mathrm{Cov}[\gamma] = c\mathrm{E_{min}}^n T^{-2} \tag{24}$$

Table 1. C and n for different distributions

Distribution	C	n
Positive uniform	4/3	2
Abs of normal	$\pi/2$	2
χ^2	2	1
Poisson	1	1

4 Evaluation

This evaluation has been done within the *MATRIS*[1] project. The goal of this project is to develop a real time system for pose estimation of cameras. One

[1] http://www.ist-matris.org

central part of this system is efficient algorithms for patch tracking in video sequences. Within the system, the result from the patch tracking is merged with data from an *Inertial Measurement unit* (IMU). To be able to combine the information in an efficient way a covariance estimate is needed for each patch.

There is a number of problems with an evaluation of tracking algorithms. The most important problem is probably to generate the ground truth without bias. In this evaluation, we solve this problem by using a synthetic image sequence generated from real textures. With this method it is possible to generate the ground truth and to simulate illumination changes.

To generate the test images a number of tools used or developed in the *MATRIS* project was used. The test data was created using this procedure:

1. Create a textured 3d-model consisting of planar patches from a number of real images.
2. Render a sequence of images showing the model from different poses. Save the 2D center position for the planar patches together with the image.
3. Start to create patches and predict the position for patches in all images. This is done with software developed in the project.
4. Estimate the start pose for the camera, the position of the camera for the first image using the image content.
5. Warp all patches that are visible from the estimated pose using a homography and save these patches.
6. Track the position for all visible patches and use this to improve the pose, combining the 3D-model and patch positions.
7. Iterate step 5-6 for all images.

This gives a number of images together with almost correctly transformed patches and their correct position. These images are then modified to simulate noise. The data has previously been used for evaluation of tracking algorithms [1].

The purpose of the evaluation is to compare the suggested covariance estimation method with an L_1 based tracking algorithm. The evaluation uses the dc-invariant tracking algorithm with subpixel accuracy [1]. To do this one condition for covariances has been derived from the definition (5), that

$$E[(x-\overline{x})^T C^{-1}(x-\overline{x})] = dim(x) \ . \tag{25}$$

This condition is used because it is simple to evaluate. Showing that a method satisfies this condition is probably good enough for practical situations, we should however note that this is not a proof that the estimated covariance is correct.

To simulate noise in the camera a number of different models are available. For this evaluation, additive Gaussian noise was used. The pixels were in the interval [0:255] and Gaussian noise with σ between 0 and 20 was added. The most interesting part with the evaluation is to compare the result for (25) with different amount of noise. Accuracy of the tracking decreases when the amount of noise increases. The experiment evaluates whether the covariance estimate will increase with the same speed.

5 Results

The covariance estimation (24) has two parameters, C and n. Most important of these parameters is n, which significantly influences the whole result whereas C "only" scales the result. The evaluation showed that $n = 1$ gave best result and is therefore used for the following results. A scaling using $C = 1$ is used, which corresponds to an assumption that the error has a Poisson distribution.

Figure 3 shows the RMS error from the tracking and the square root of (25) which is the error scaled by the estimated covariance. Most important is the shape of the curves and the RMS error measured in pixels is therefore scaled to simplify the evaluation. Originally the RMS tracking error started at ≈ 0.15. The figure shows that the normalized average error is closer to a constant function than the original tracking error, especially for small amounts of noise, $\sigma < 10$. The figure shows a trend, the estimated covariance underestimates the increase of the error in the tracking with high noise levels. Whether the covariance estimate is too optimistic or performance of the block matching could be improved with high noise levels is still an open question.

We can also see that the scaling of the covariance is slightly wrong, the graph does not start at 2, which is the dimension of the estimated parameter. To be able to use the covariance in combination with other sensors, this scaling has to be manually adjusted.

The results show that the suggested method is significantly better than using no covariance at all and the estimation of the covariance is very fast.

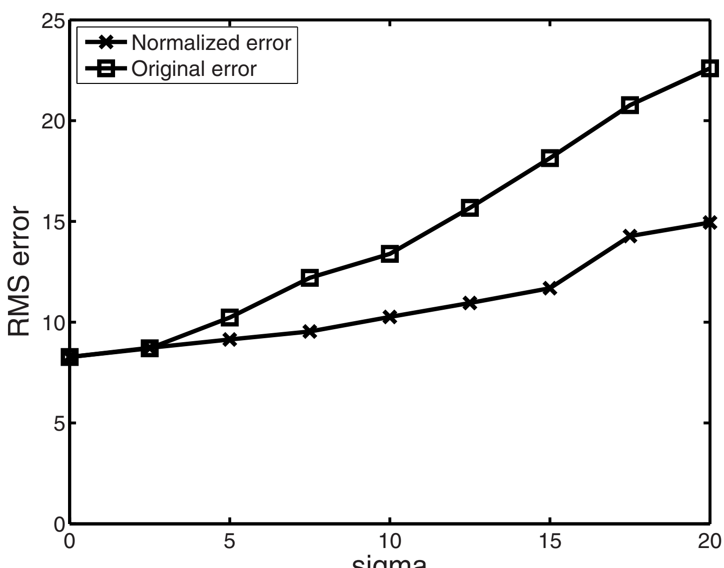

Fig. 3. Normalized error vs (scaled) original error

6 Conclusions

In this paper we proposed a method for estimating the covariance matrix of SAD block matching. An algorithm for computing the covariance using dual frames has been formulated. This provides an efficient method to calculate the covariance.

In the second part of the paper, an evaluation of the proposed method for covariance estimation has been performed. In the evaluation a dc-invariant SAD block matching method with subpixel interpolation was used. The evaluation showed that the suggested method for covariance estimation was significantly better than assuming that each patch has the same error. The calculational complexity of the suggested method is low and it can therefore be applied, with almost unchanged computational complexity.

Acknowledgment

This work has been supported by EC Grant IST-2002-002013 MATRIS.

References

1. Skoglund, J., Felsberg, M.: Evaluation of subpixel tracking algorithms. In: ISVC (2), pp. 374–382 (2006)
2. Dougherty, E.R.: Random Processes for Image and Signal Processing. SPIE press (1999)
3. Ljung, L.: Sytem Identifiaction. Prentice hall, Englewood Cliffs (1999)
4. Kanazawa, Y., Kanatani, K.: Do we really have to consider covariance matrices for image features? ICCV 02, 301 (2001)
5. Fessler, J.A.: Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): applications to tomography. IEEE Tr. Im. Proc. 5(3), 493–506 (1996)
6. Sun, Q., DeJong, G.: Feature kernel functions: Improving SVMs using high-level knowledge. In: CVPR (2). pp. 177–183 (2005)
7. Granlund, G., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers, Dordrecht (1995)