Template Matching using Fast Normalized Cross Correlation

Kai Briechle and Uwe D. Hanebeck

Institute of Automatic Control Engineering, Technische Universität München,
80290 München, Germany

ABSTRACT

In this paper, we present an algorithm for fast calculation of the normalized cross correlation (NCC) and its application to the problem of template matching. Given a template t, whose position is to be determined in an image f, the basic idea of the algorithm is to represent the template, for which the normalized cross correlation is calculated, as a sum of rectangular basis functions. Then the correlation is calculated for each basis function instead of the whole template. The result of the correlation of the template t and the image f is obtained as the weighted sum of the correlation functions of the basis functions.

Depending on the approximation, the algorithm can by far outperform Fourier–transform based implementations of the normalized cross correlation algorithm and it is especially suited to problems, where many different templates are to be found in the same image f.

Keywords: Normalized cross correlation, image processing, template matching, basis functions

1. INTRODUCTION

A basic problem that often occurs image processing is to determine the position of a given pattern in an image, or part of an image, the so–called region of interest. This problem is closely related to the determination of a received digital signal in signal processing using e.g. a matched filter.

Two basic cases can be differentiated:

- The position of the pattern is unknown
- An estimate for the position of the pattern is given

Usually, both cases have to be treated to solve the problem of determining the position of a given pattern in an image. In the latter case, the information about the position of the pattern can be used to reduce the computational effort significantly. It is also known as feature tracking in a sequence of images.[1,2]

For both feature tracking and the initial estimation of the position of the given pattern, a lot of different, well–known algorithms have been developped.[3,4] One basic approach that can be used in both cases mentioned above, is template matching. This means that the position of the given pattern is determined by a pixel-wise comparision of the image with a given template, that contains the desired pattern. For this, the template is shifted u discrete steps in the x direction and v steps in the y direction of the image, and then the comparison is calculated over the template area for each position (u, v). To calculate this comparison, normalized cross correlation is a reasonable choice in many cases.[5] Nevertheless, it is computationally expensive and therefore a fast correlation algorithm that requires less calculations than the basic version is of interest.

In section 2, the problem treated in this paper is defined and a brief summary of the normalized cross correlation algorithm is given. Section 3 introduces a new, fast algorithm that computes the normalized cross correlation in an efficient manner for an approximation of the template. The performance of the new algorithm is compared to standard naive implementation of the normalized cross correlation and to the well–known Fourier–transform. Section 4 briefly describes, how the algorithm can be applied recursively. In section 5, an example is presented, in which the proposed algorithm is applied for template matching. Finally an outlook to future research activities is presented.

K. Briechle: E-mail: kai.briechle@ei.tum.de
U. D. Hanebeck: E-mail: uwe.hanebeck@ei.tum.de

2. NCC–ALGORITHM

The problem treated in this paper is to determine the position of a given pattern in a two dimensional image f. Let $f(x,y)$ denote the intensity value of the image f of the size $M_x \times M_y$ at the point $(x,y), x \in \{0,\ldots,M_x-1\}$, $y \in \{0,\ldots,M_y-1\}$. The pattern is represented by a given template t of the size $N_x \times N_y$. A common way to calculate the position (u_{pos}, v_{pos}) of the pattern in the image f is to evaluate the normalized cross correlation value γ at each point (u,v) for f and the template t, which has been shifted by u steps in the x direction and by v steps in the y direction. Equation (1) gives a basic defintion for the normalized cross correlation coefficient.

$$\gamma = \frac{\sum_{x,y}(f(x,y)-\bar{f}_{u,v})(t(x-u,y-v)-\bar{t})}{\sqrt{\sum_{x,y}(f(x,y)-\bar{f}_{u,v})^2 \sum_{x,y}(t(x-u,y-v)-\bar{t})^2}} \tag{1}$$

In (1) $\bar{f}_{u,v}$ denotes the mean value of $f(x,y)$ within the area of the template t shifted to (u,v) which is calculated by

$$\bar{f}_{u,v} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x,y). \tag{2}$$

With similar notation $\bar{t}$ is the mean value of the template t. The denominator in (1) is the variance of the zero mean image function $f(x,y)-\bar{f}_{u,v}$ and the shifted zero mean template function $t(x-u,y-v)-\bar{t}$. Due to this normalization, $\gamma(u,v)$ is independent to changes in brightness or contrast of the image, which are related to the mean value and the standard deviation.

The desired position (u_{pos}, v_{pos}) of the pattern, which is represented by t, is equivalent to the position (u_{max}, v_{max}) of the maximum value γ_{max} of $\gamma(u,v)$. Due to the normalization, the use of (1) for the calculation of the position of the pattern is more robust than other similarity measures, like simple covariance or the sum of the absolute differences (SAD). Nevertheless the main drawback is, that the calculation of (1) is computationally expensive. For the denominator, which normalizes the cross correlation coefficient, at every point $(u,v), u \in \{0,\ldots,M_x-N_x\}$, $v \in \{0,\ldots,M_y-N_y\}$ of the image, at which $\gamma(u,v)$ is determined, the energy of the zero mean image

$$e_f(u,v) = \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} (f(x,y)-\bar{f}_{u,v})^2 \tag{3}$$

and the mean of the image within the area of the template function $\bar{f}_{u,v}$ (2) have to be recalculated. If this calculation is implemented in a straightforward naive way, according to (1), the number of calculations is proportional to $N_x N_y (M_x-N_x)(M_y-N_y)$, though the energy of the zero mean template function

$$e_t(u,v) = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (t(x,y)-\bar{t})^2 \tag{4}$$

and the mean of the template function $\bar{t}$ have to be precalculated only once. This computational effort is not acceptable for most practical applications. The nominator in (1) can be calculated in the frequency range using the well–known Fourier–transform, yet the number of calculations is still comparativly high, and an algorithm that calculates the normalized cross correlation with less calculations is of great interest.

To overcome these complexity problems, an efficient method to calculate the denominator of the normalized cross correlation coefficient is proposed by Lewis.[5] The main idea is to precalculate sum–tables containing the integral over the image function $f(x,y)$ and the squared image function $f^2(x,y)$ (running sum) once for each image f, and use these tables for efficient calculation of the expression $(f(x,y)-\bar{f}_{u,v})^2$ at each point (u,v), at which the normalized cross correlation coefficient is evaluated. Using these sum–tables, the resulting number of calculations for the denominator does no longer depend on the size of the template N_x, N_y but only on the size of the image function M_x, M_y. A brief description of this calculation is given in section 3.1.

In section 3.2 the key idea, that allows a very efficient calculation of the numerator of (1) is explained in detail. Thus, the normalized cross correlation coefficient (1) can be calculated for an approximated template function $\tilde{t}(x,y)$ with an order of magnitude less calculations than the standard FFT approach, which opens up many new applications.

3. FAST NCC–ALGORITHM

3.1. Calculation of the denominator

To simplify the calculation of the denominator of the normalized cross correlation coefficient, the key idea is to use two sum tables $s(u,v)$ and $s^2(u,v)$ over the image function $f(x,y)$ and the image energy $f^2(x,y)$.[5] The sum table of the image function is recursively defined by

$$s(u,v) = f(u,v) + s(u-1,v) + s(u,v-1) - s(u-1,v-1). \tag{5}$$

A similar recursive definition for the sum table over the image energy is given by

$$s^2(u,v) = f^2(u,v) + s^2(u-1,v) + s^2(u,v-1) - s^2(u-1,v-1) \tag{6}$$

with $s(u,v) = s^2(u,v) = 0$ when either $u, v < 0$. The following algorithm for simplified calculation of the denominator can be applied to the whole image function or to any subimage of the image function $f(x,y)$. Then the sum tables have to calculated only for this subimage region. With these tables, (2) can be calculated in a very efficient manner, independent of the size N_x, N_y of the template

$$\sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x,y) = s(u+N_x-1, v+N_y-1) - s(u-1, v+N_y-1) - s(u+N_x-1, v-1) + s(u-1, v-1). \tag{7}$$

It can be seen from (7), that only three additions / subtractions are necessary to evaluate the double sum over $f(x,y)$ by evaluation of the sum–table $s(u,v)$.

The denominator of (1) is then evaluated using the running sum tables (5), (6) and

$$\sum_x \sum_y (f(x,y) - \bar{f}_{u,v})^2 = \sum_x \sum_y f^2(x,y) - 2\bar{f}_{u,v} \sum_x \sum_y f(x,y) + \sum_x \sum_y \bar{f}_{u,v}^2. \tag{8}$$

In (8) and all equations in the remainder of this paragraph, the double sum $\sum_x \sum_y$ is evalutated over the region of the template, which means $u < x < u + N_x - 1$ and $v < y < v + N_y - 1$. With

$$\sum_x \sum_y \bar{f}_{u,v}^2 = N_x N_y \left(\frac{1}{N_x N_y} \sum_x \sum_y f(x,y) \right)^2 \tag{9}$$

(8) can be simplified to

$$\sum_x \sum_y (f(x,y) - \bar{f}_{u,v})^2 = \sum_x \sum_y f^2(x,y) - \frac{1}{N_x N_y} \left(\sum_x \sum_y f(x,y) \right)^2. \tag{10}$$

The sum expressions in (10) over $f^2(x,y)$ and $f(x,y)$ can efficiently be calculated using the running sum tables (5), (6). Moreover, one square root has to calculated for each point (u,v) to determine the denominator of (1).[5]

3.2. Calculation of the numerator

Application of the algorithm presented in the last subsection allows efficient calculation of the denominator, but the number of computations required to calculate the numerator of the NCC–coefficient (1) is still comparatively high, even if it is done in the frequency range with an FFT algorithm. Therefore, further simplification of this calculation is required. The numerator can be rewritten as

$$N(u,v) = \sum_x \sum_y f(x,y) t'(x-u, y-v) - \bar{f}_{u,v} \sum_x \sum_y t'(x-u, y-v) \tag{11}$$

where $t'(x-u, y-v)$ is the zero mean template function defined by

$$t'(x-u, y-v) = t(x-u, y-v) - \bar{t}. \tag{12}$$

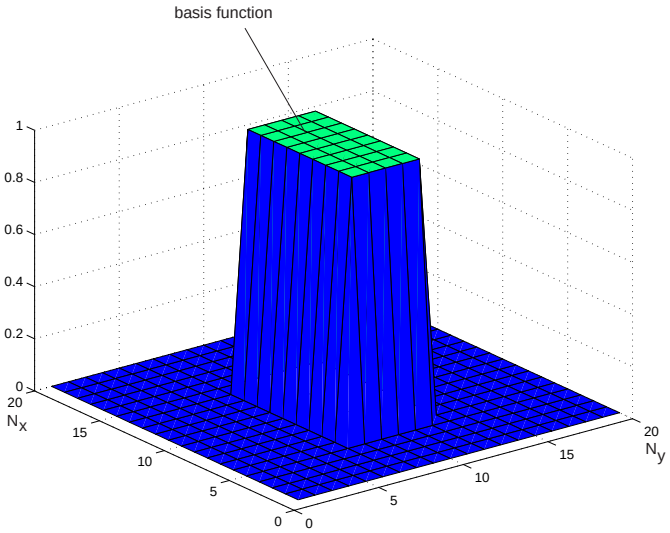


Figure 1. A single rectangular basis function $t_i(x, y)$.

As $t'(x, y)$ has zero mean and thus also zero sum, the term $\bar{f}_{u,v} \sum_x \sum_y t'(x-u, y-v)$ is zero as well,[5] and the numerator of (1) can written as

$$N(u, v) = \sum_x \sum_y f(x, y) t'(x-u, y-v). \tag{13}$$

The *basic idea* to simplify the calculation of the numerator is to expand the zero mean template function $t'(x, y)$ to the weighted sum of K rectangular basis functions t_i, yielding an approximation $\tilde{t}(x, y)$ of the template function

$$\tilde{t}(x, y) = \sum_{i=1}^{K} k_i t_i(x, y). \tag{14}$$

The numerator of the normalized cross correlation coefficient can then be efficiently evaluated with this approximation. The basis function $t_i(x, y)$ is constant equal 1 inside an rectangular area $x_l^i \leq x \leq x_u^i \wedge y_l^i \leq y \leq y_u^i$ and zero otherwise. Thus x_l^i and x_u^i are the indices of the lower and the upper bound of the rectangular area that is constant equal 1 in the x–direction. y_l^i and y_u^i are the respective indices in the y–direction. Figure 1 shows an example for a rectangular basis function $t_i(x, y)$ where $N_x = N_y = 20$. It is constant equal one for $6 \leq x \leq 14 \wedge 8 \leq y \leq 12$. The quality of the approximation $\tilde{t}(x, y)$ of the original zero mean template function $t'(x, y)$ depends both on the choice of the basis functions, that means, on their lower and upper bounds x_l^i, x_u^i and y_l^i, y_u^i and on the number of basis functions that are used.

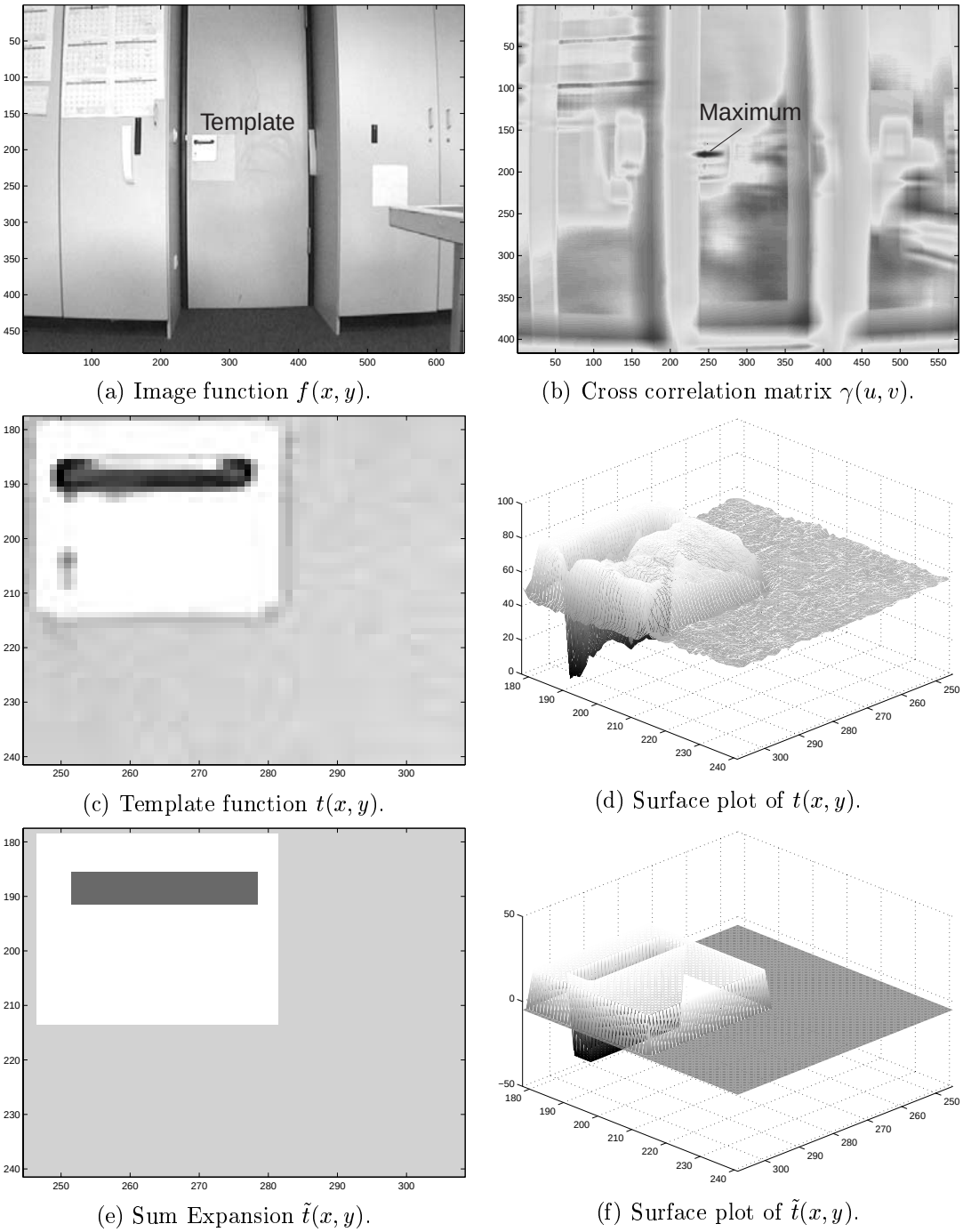
Using the sum expansion of the template function $\tilde{t}(x, y)$ allows to calculate an approximation for the numerator of the cross correlation coefficient

$$\tilde{N}(u, v) = \sum_{i=1}^{K} k_i \sum_{x=x_l^i+u}^{x_u^i+u} \sum_{y=y_l^i+v}^{y_u^i+v} f(x, y), \tag{15}$$

where K is the number of basis functions and k_i is the coefficient for basis function i. Remember that x_l^i and y_l^i denote the indices of the lower limits of basic function i and x_u^i and y_u^i subsequently are the indices of the upper limits. (15) is obtained from (13) using (14), because the basis functions $t_i(x, y)$ are either constant one or zero. With the running sum table over the image function (5), that has already been calculated to simplify the determination of the denominator, which cannot be calculated in the frequency range, the inner double sum in (15) can be calculated

with only 3 addition/subtraction operations.

$$\sum_{x=x_l^i}^{x_u^i} \sum_{y=y_l^i}^{y_u^i} f(x,y) = s(x_u^i, y_u^i) - s(x_u^i, y_l^i - 1) - s(x_l^i - 1, y_u^i) + s(x_l^i - 1, y_l^i - 1). \tag{16}$$



(a) Image function $f(x, y)$.

(b) Cross correlation matrix $\gamma(u, v)$.

(c) Template function $t(x, y)$.

(d) Surface plot of $t(x, y)$.

(e) Sum Expansion $\tilde{t}(x, y)$.

(f) Surface plot of $\tilde{t}(x, y)$.

Figure 2. Experiment: Template matching with fast normalized cross correlation.

This means, that the number of calculations required to determine an approximation for the numerator in (1) depends only on the number of rectangular basis functions used, but not on their size. The approximated cross

correlation coefficient $\tilde{\gamma}(u,v)$ is

$$\tilde{\gamma}(u,v) = \frac{\tilde{N}(u,v)}{\sqrt{\sum_{x,y}(f(x,y)-\bar{f}_{u,v})^2 \sum_{x,y}(t(x-u,y-v)-\bar{t})^2}}, \tag{17}$$

whose denominator is equivalent to (1) and can be efficiently calculated using the sum–tables (5), (6). Substituting (15) and (16) finally yields the following compact formula for the approximated cross correlation coefficient

$$\tilde{\gamma}(u,v) = \frac{\sum_{i=1}^{K} k_i \left(s(x_u^i+u, y_u^i+v) - s(x_u^i+u, y_l^i+v-1) - s(x_l^i+u-1, y_u^i+v) + s(x_l^i+u-1, y_l^i+v-1)\right)}{\sqrt{\sum_{x,y}(f(x,y)-\bar{f}_{u,v})^2 \sum_{x,y}(t(x-u,y-v)-\bar{t})^2}}. \tag{18}$$

3.3. Analysis of complexity for the calculation of the numerator

The number of computations required to calculate the numerator of the NCC coefficient for an image of the size M_x*M_y and a template of the size N_x*N_y that has been approximated with K rectangular basis functions is given in table 1. It can be seen, that the number of multiplications depends linearly on the number of basis functions K for a given image function $f(x,y)$. For each point (u,v), at which the normalized cross correlation coefficient $\gamma(u,v)$ is evaluated, K multiplications are required to calculate the numerator, one for each basis function.

In contrast to this, a direct evaluation of $\gamma(u,v)$ requires $N_x * N_y$ multiplications at each point (u,v) at which γ is evaluated. The complexity of the FFT depends on the size of the template and the image function.[6] When M is much larger than N it may even exceed the number of computations required by the direct method. Independent of M and N it can also by far exceed the number of computations required by the proposed algorithm depending on the number of basis functions K.

The number of computations required to set up the sum–tables and evaluate the denominator of the normalized cross correlation coefficient is not regarded in table 1 and table 2. To calculate the denominator, that normalizes the cross correlation coefficient, the method proposed by Lewis[5] is far more efficient than a naive, direct calculation. Thus the computations to set up the sum–tables are necessary for all three methods dicussed here and are not included in the comparison. Of course, the proposed algorithm for calculation of the nominator can also be applied to calculate efficiently an ordinary cross correlation without normalization, but then the extra effort to set up the sum tables has to be taken into account compared to the standard FFT, that does not require any sum–tables.

	Add / Sub	Mult
Direct calc.	$N_xN_y(M_x-N_x+1)(M_y-N_y+1)$	$N_xN_y(M_x-N_x+1)(M_y-N_y+1)$
FFT	$9M_xM_y\log_2(M_xM_y)$	$6M_xM_y\log_2(M_xM_y)$
New alg.	$(4K-1)(M_x-N_x+1)(M_y-N_y+1)$	$K(M_x-N_x+1)(M_y-N_y+1)$

Table 1. Analysis of Complexity, numerator only.

$M_x = 640$, $M_y = 480$, $N_x = N_y = 64$, $K = 3$	Add / Sub	Mult
Direct calc.	985.5 Mio	985.5 Mio
FFT	50.4 Mio	33.6 Mio
New alg.	2.65 Mio	0.72 Mio

Table 2. Analysis of Complexity, Example.

3.4. Determination of basis functions

To apply the proposed algorithm for efficient calculation of the numerator of (1), a set of basis functions t_i has to be determined to approximate the original template function. For simple template functions $t(x,y)$, a set of basis

functions can be determined manually, as shown in section 5. For automatic determination of the basis functions, the quadratic criterion

$$J = \sum_{x,y} (t'(x,y) - \tilde{t}(x,y))^2 \tag{19}$$

is used to assess the quality of the approximation. A recursive algorithm divides the template function $t(x,y)$ into rectangular basis functions. It starts with a single basis function $t_1(x,y) = \bar{t} = constant$ and calculates J using (14) and (19). If $J > J_{max}$, where J_{max} is a predefined threshold, the basis function is devided into two basis functions and the coefficients $k_i, i = 1,2$ are recalculated under the condition, that J is minimized. This process is continued recursively for each basis function t_i until $J < J_{max}$. Note, however, that the approximation found by this algorithm is not globally optimal with regard to the number of basis functions required to approximate the template. This means, that a different approximation might have the same qualitiy J, but uses less than K basis functions t_i. Many features encountered in an indoor environment can nevertheless be well approximated with a few basis functions by means of the recursive algorithm. The number of computations required to calculate the approximated numerator of (1) is directly proportional to the number of basis functions K that are used for the approximation. Therefore, a small number of well–chosen basis functions is desirable. On the other hand, using to few basis functions results in a bad approximation of the template and thus of the numerator (1). Automatic determination of the basis functions allows to determine a trade–off between complexity of the calculation and the approximation error.

4. RECURSIVE APPLICATION OF THE ALGORITHM

As pointed out in the last section, the qualitiy of the approximation of the template directly effects the result of the proposed algorithm, the approximated numerator of (1). The results of the direct calculation and the proposed algorithm are the same, if $K = N_x * N_y$ basis functions are used to represent the template. Nevertheless this is not practical, as the computational complexity of the proposed algorithm converges to that of a direct calculation of $\gamma(u,v)$ and thus exceeds the complexity of the FFT based calculation.

As long as the approximation of the template function is sufficiently good, the pattern represented by $t(x,y)$ is robustly detected using the maximum value of $\tilde{\gamma}(u,v)$ with very little computational effort, as shown in the expample in chapter 5. If the approximation of the template $t(x,y)$ calculated by the algorithm proposed in subsection 3.4 is not good enough to yield a sufficiently good approximation of the normalized cross correlation coefficient $\gamma(u,v)$, the proposed algorithm is applied recursively. The definition of a sufficiently good approximation depends on the given problem. In template matching applications, it is necessary to determine the position of the template in an image by searching the maximum of $\gamma(u,v)$. In this context, a good approximation means that the maximum of $\tilde{\gamma}(u,v)$ is equal or close to the maximum of $\gamma(u,v)$.

For a recursive application of the proposed algorithm, the cross correlation coefficient $\tilde{\gamma}(u,v)$ is calculated with approximations of the template function $t(x,y)$ that use an increasing number of basis functions in each step of the recursion. In the first step, $\tilde{\gamma}(u,v)$ is calculated with a very rough approximation of $t(x,y)$ using only a few basis functions. The maximum error in $\tilde{\gamma}(u,v)$ obtained by this approximation can be estimated. This error bound is then used to determine a subset $f_2(x,y)$ of all pixels of the image function $f(x,y)$. For this subset, the proposed algorithm is applied recursively with a better approximation of $t(x,y)$, yielding a smaller subset $f_3(x,y)$. The process stops when the number of pixels in the subset $f_N(x,y)$ of $f(x,y)$ is sufficiently small. The correct maximum value of $\gamma(u,v)$ can then be found by direct evaluation of (1) on subset $f_N(x,y)$, which is equivalent to using a representation with the maximum number of basis functions $K = N_x * N_y$ and the proposed algorithm.

5. EXAMPLE

Figure 2a) shows a normal camera image taken from a typical indoor environment. The handle of the door is the pattern to be found within this image. The template function $t(x,y)$ of the pattern, which has the size 64×64 is displayed magnified in Fig. 2c). This template can well be approximated by the weighted sum of 3 rectangular basis functions, which yields a new template function $\tilde{t}(x,y)$ (Fig. 2e). The basis functions would normally be calculated with the proposed recursive algorithm, but this would lead to a worse approximation using more basis function. Therefore for this example, the three basis functions t_i were selected manually to demonstrate how the cross correlation algorithm works. A similar, but less obvious result can be obtained with a worse approximation, that is automatically computed. Note that the approximated template function $\tilde{t}(x,y)$ has zero mean. Figure 2d)

and e) show a surface plot of the original and the approximated template function $t(x,y)$ and $\tilde{t}(x,y)$. The height of the plot corresponds to the value of the function $t(x,y)$ and $\tilde{t}(x,y)$ at the point (x,y).

Note, however, that the values $-5, 18$ and -30 of $\tilde{t}(x,y)$ are not equal to the coefficients k_i of the $K = 3$ basis functions, because the basis functions $t_i(x,y)$ overlap in this example. This means, that the coefficient of the smallest rectangle is calculated taking into account the two outer rectangles, as the actual value of $\tilde{t}(x,y)$ in the area of the smallest rectangle is equal to the weighted sum of all three basis functions.

In Fig. 2b) the resulting cross correlation function $\tilde{\gamma}(u,v)$ of the NCC computed with the algorithm (9) (10) is given. Dark pixels correspond to high values of $\tilde{\gamma}$ that are close to one, and light pixels to low values close to -1. Despite the rough approximation of the template function (Fig. 2e)), the fast NCC–algorithm determines the position of the template in the original image correctly, yielding the maximum value of $\tilde{\gamma}(u,v)$ at $[x,y]' = [245\ 178]'$.

5.1. Efficiency

For the example that uses a VGA-camera image, the size of the image function is 640×480 pixel and 64×64 pixels for the template function. Table 2 shows, that the number of multiplications for the numerator is reduced 47 times compared to the FFT and 2048 times compared to a direct calculation, assuming that the sum tables used for calculating the denominator are required for each algorithm. This means, that up to 140 basis functions may be used, before the computational load is equivalent to the FFT algorithm. It is assumed that the FFT algorithm requires that f and t be extended with zeros to a common power of two (zero padding).

6. CONCLUSIONS

A new fast algorithm for the computation of the normalized cross correlation has been derived, that uses a sum expansion of the given template function t and rectangular basis functions $t_i(x,y)$. The number of calculations required to evaluate the normalized cross correlation coefficient $\gamma(u,v)$ for the image function $f(x,y)$ depends linearly on the number of basis functions used, but not on the size of the template t. It has been demonstrated, that the position of a simple feature like a door handle can be determined in a VGA camera image with a 47 times less multiplications compared to an evaluation of $\gamma(u,v)$ that uses the FFT. This makes the proposed algorithm attractive for real time image processing applications like feature tracking.

ACKNOWLEDGMENTS

This research has been sponsored by the Bavarian Research Foundation (BFS) as part of the project "DIROKOL".

REFERENCES

1. Gregory D. Hager and Kentaro Toyama, *The XVision System: A General–Purpose Substrate for Portable Real-Time Vision Applications* Computer Vision and Image Understanding, Volume 69, Number 1, pp. 23-37, 1998.
2. B. D. Lucas and T. Kanade, *An Iterative Image Registration Technique with an Application to Stereo Vision*, IJCAI 1981.
3. Hirochika Inoue, Tetsuya Tachikawa and Masayuki Inaba, *Robot Vision System with a Correlation Chip for Real–time Tracking, Optical Flow and Depth Map Generation*, Proceedings of the IEEE International Conference on Robotics and Automation, pp 1621-1626, France, 1992.
4. Changming Sun, *A Fast Stereo Matching Method*, Digital Image Computing: Techniques and Applications, pp. 95-100, Massey University, Auckland, New Zealand, 10-12 December 1997.
5. J. P. Lewis, *Fast Normalized Cross–Correlation*, Industrial Light and Magic.
6. Rafael C. Gonzales, Paul Wintz, *Digital Image Processing*, Second Edition, Addison–Wesley Publishing Company, Reading, Massachusetts, November 1987, ISBN 0-201-11026-1.