A Sign Bit Only Phase Normalization for Rotation and Scale Invariant Template Matching

Meng Ma[1], Arjan van Genderen[1] and Peter Beukelman[2]

[1]Computer Engineering, EEMCS, TU Delft, the Netherlands,Mekelweg 4(15th floor), 2628 CD Delft,The Netherlands

Phone: +31 (0)15 2786 217

Fax: +31 (0)15 2784 898

Email: {m.ma|a.j.vangenderen}@ewi.tudelft.nl

[2]Eonic B.V.,Delftechpark 26,2628 XH, Delft, the Netherlands

Phone: +31(0)15 2600 432

Fax: +31(0)15 2600 431

Email: peter.beukelman@eonic.com

Abstract— **Using Symmetric Phase Only Matched Filter (SPOMF) to locate the position of a known object in static pictures or real-time streaming pictures has been proved to be a reliable method. A key issue of SPOMF is the normalization of the phase of the complex matrix. In this paper, several different solutions for phase normalization are introduced and compared. Starting from the CORDIC solution, experimental results indicate that phase angle quantization shows little influence to the quality of the phase normalization but can greatly increase the performance. The method is to take some most significant bits of the real part and the imaginary part of the complex numbers to represent the phase angles. Based on the MatLab simulation results, a highly efficient solution that is called Sign Bit Only Phase Normalization is developed. This solution only uses the sign bit to represent the phase angles so that all the complex numbers can only have 4 possible phase angles. In that case, the Sign Bit Only solution - while being much easier to implement - still shows acceptable results, in which the peak magnitude in the correlated image is decreased to about 90%. A mathematical explanation that proves the reliability of this solution and experimental results are also given in this paper.**

Keywords— **Image Registration; FFT; SPOMF; Phase Normalization**

I. INTRODUCTION

Symmetric Phase Only Matched Filter (SPOMF) has been proved to be a reliable method for detecting a known object in an image.

The method is that both the template image and search image will be transformed to the frequency domain. In the frequency domain, the spectral components of search image and template image are element wise conjugated multiplied. This operation actually calculates the phase differences between spectral points in the search image and the corresponding pixel in the template image. All the vectors in the resulting matrix of the multiplication will be normalized to 1. Then, the resulting matrix will be converted back to the spatial domain and the peaks that exceed the threshold indicate the found location of this object in the search image [2].

SPOMF works well for exact matches even when noise is present or only part of the object is visible. It fails when the match cannot be obtained by a translation and scaling and rotation are added. In that case, the rotation angle and scaling factor should be first detected and then the template image can be compensated for. After that, regular SPOMF can be used to detect the location(s) of object(s).

This paper focuses on the development of a fast phase normalization method using only the sign bit of the numbers in the spectrum. Experiment results show that this method can provide acceptable performance and can be very easy to implement.

II. BACKGROUND

A. Theory

In [3], B. Srinivasa Reddy and B. N. Chatterji presented a FFT based method to detect translation, rotation and scale.

- **Translation**

The translation of two images in the spatial domain is represented only by phase difference in the frequency domain. Let f_1 and f_2 be the two images that differ by a displacement of (x_0, y_0) , the two images are related like this:

$$f_2(x,y) = f_1(x - x_0, y - y_0)$$

Their Fourier transform will then be related like this:

$$F_2(\xi,\eta) = e^{-j2\pi(\xi x_0+\eta y_0)} \bullet F_1(\xi,\eta)$$

The phase difference can be calculated by the elemental wise multiplication of the spectrum of the search image and the complex conjugated spectrum of the template image.

- **Rotation**

Rotation in the spatial domain still results in a rotation in the frequency domain. Let $R(\theta)$ be the rotation in the spatial domain. And assume the translation still remains as (x_0, y_0), the two images are related as:

$$f_2(x, y) = f_1(x\cos\theta + y\sin\theta - x_0, -x\sin\theta + y\cos\theta - y_0)$$

After being transformed to Fourier domain, we obtain:

$$F_2(\xi, \eta) = e^{-j2\pi(\xi x_0 + \eta y_0)} \bullet F_1(\xi\cos\theta + \eta\sin\theta, -\xi\sin\theta + \eta\cos\theta)$$

Notice that the spectrum of f_2 is actually the rotated spectrum of f_1 if the phase is taken away.

- **Scale**

Scale in the spatial domain remains an inverse scale in the frequency domain. Scale factor(s) can be calculated by converting the coordinate system of the frequency domain to logarithmic scale. Let f_1 be the scaled replica of f_2 with scale factor (a, b) which stands for horizontally scale and vertically scale. The relation of f_1 and f_2 in the frequency domain is like:

$$F_2(\xi, \eta) = \frac{1}{|ab|} F_1(\xi/a, \eta/b)$$

Once the coordinate system is converted to logarithmic scale, scaling can be reduced to translation that can be calculated by regular SPOM. If we ignore the multiplication constant $1/ab$, we get:

$$F_2(\log\xi, \log\eta) = F_1(\log\xi - \log a, \log\eta - \log b)$$

In that case, the scale factors can be easily obtained.

B. Reconfigurable Platform

Based on the theory, a rotation and scale invariant template matching algorithm can be developed. The search image and the template image will firstly be converted to the frequency domain and mapped to the log-polar coordinate system to detect the rotation and scale factors. Afterwards, rotating and scaling the template image will compensate the rotation and scale factors. Then, the regular SPOMF can be applied to detect the actual location of the target object.

A polar FFT method was presented in [4] to directly calculate the fourier spectrum under polar coordinates.

The whole algorithm is implemented to a platform with four PowerFFT boards. The PowerFFT board includes a PowerFFT processor that has a 57 bit datapath (9 bits exponent and 24 bits mantissa for real and imaginary number) and two FPGAs. See figure 1.

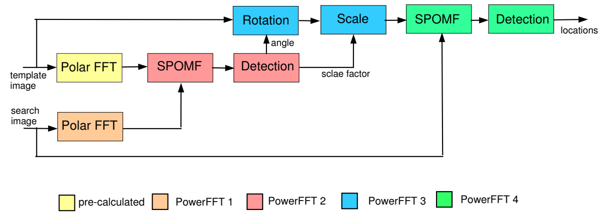


Fig. 1. The reconfigurable platform

III. PHASE NORMALIZATION

The SPOMF algorithm requires the magnitude of the spectrum to be 1 after conjugated multiplying the spectrums of search image and template image. We have developed a simple but efficient method to normalize the magnitude of the spectrum.

A. A. First Idea: CORDIC

An advantage of CORDIC over other solutions is that it only uses shifts and additions. It can be efficiently implemented on hardware without complex units. Besides, each rotation step in CORDIC is quite similar so that they can be pipelined to reach a high performance.

CORDIC works by rotating the coordinate system through constant angles until the angle is reduced to zero. The angle offsets are selected such that the operations on x and y are only shifts and adds [5]. This method can be used here to first rotate the vector to x-axis, normalize to 1 and then rotate back to the original angle. See figure 2

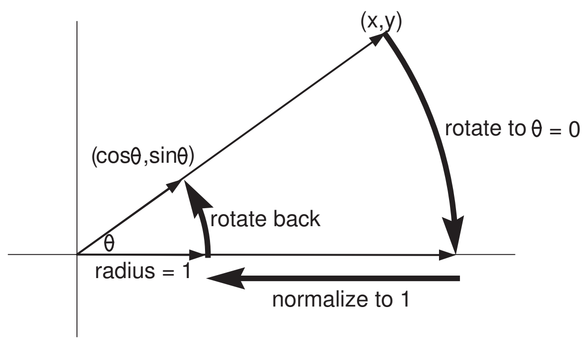


Fig. 2. Phase normalization with CORDIC

To rotate a complex number to the real axis, we can use a simplified version of CORDIC. Because the rotation direction coefficients depend on the sign of imaginary part, the calculation of θ is not necessary. Since we also want to rotate the vector back to the original angle, the rotation direction coefficients can be reused (notice that the coefficients should be inversed before using) in the rotating back operation. A simple model of CORDIC solution is given in the figure 3.

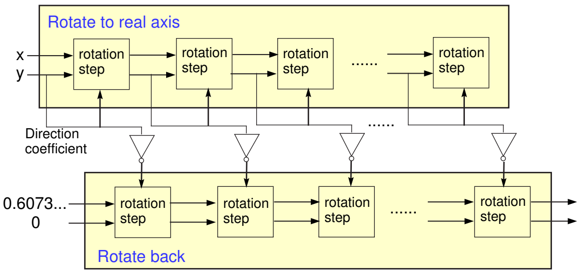


Fig. 3. Phase normalization with CORDIC

The input of the rotate back unit is always the same for every vector. Notice that the real part of the initial vector is 0.6073. This is the scale factor K that has been pre-calculated. If the rotation is completed by N steps, the scale factor $K = \frac{1}{P} = \prod_{n=0}^{N} \cos(\arctan(\frac{1}{2^n}))$. When N is a large number, $P \approx 1.6468$, $K \approx 0.607253$ [6]. The rotate back unit only needs the direction coefficients from the sign bit of the imaginary part so the two rotate units can work concurrently.

The number of iteration steps for the CORDIC algorithm is basically determined by the representing bit number of the input x and y [6]. The reason of this is that we set every $tan(\theta_n)$ to 2^{-n}, meaning that every step, the $tan(\theta_n)$ becomes half of the previous value. And the curve of $tan(\theta)$ can be considered as a linear function of $f(x) = x$ in the region near zero. That results in the fact that the θ_n itself also approximately becomes half in every step. Table I gives an example of 10 bits precision to demonstrate the linear characteristic of $tan(\theta_n)$.

B. Lookup table and CORDIC hybrid solution

Another way to obtain the direction coefficients is to remember them in a lookup table since there is large memory space in the FPGA chip. The problem is that both the real part and the imaginary part of the complex number are represented by floating-point numbers with 24 bits for the mantissa and 9 bits for the common exponent. Therefore, it is impossible to establish this lookup table in full accuracy since it results in a table with 2^{48}entries. The solution is that we use only several most significant bits of x and y, while the content of the table is the direction coefficient sequence. See figure 4.

Figure 5 gives a MatLab simulation of applying phase normalization using the full CORDIC and the hybrid solution. The search image is a mig-25 fighter in a noisy background with a high brightness area in the top left corner to test the normalization. The template image is the same mig-25 located in top left corner of the image. Figure 5(c) shows the correlated image after SPOMF with the full CORDIC operation while figure 5(d) shows the correlated image with the hybrid solution which takes the 5 most significant bits of x and y. With 5 bits of x and 5 bits of y, there are $2^{10} = 1024$ entries in the table. And the coefficients sequence contains 24 bits since there are 24 rotation steps for the CORDIC. The total size of the table is then 4352 bytes. Compared to full CORDIC, the hybrid solution has no obvious weakness. The magnitude of the peak is 99.96% of the peak in the full CORDIC and the noise level barely changes. Tests for other images also give almost the same good results. Those tests show that using few bits to represent the phase still provides good detection quality. Then, does it work with even fewer bits?

i	2^{-i}	$\arctan(2^{-i})$
0	1.0000000000	0.1100100100
1	0.1000000000	0.0111011011
2	0.0100000000	0.0011111011
3	0.0010000000	0.0001111111
4	0.0001000000	0.0001000000
5	0.0000100000	0.0000100000
6	0.0000010000	0.0000010000
7	0.0000001000	0.0000001000
8	0.0000000100	0.0000000100
9	0.0000000010	0.0000000010
10	0.0000000001	0.0000000001

TABLE I

THE VALUE OF $\arctan(2^{-i})$ WITH 10-BIT PRECISION

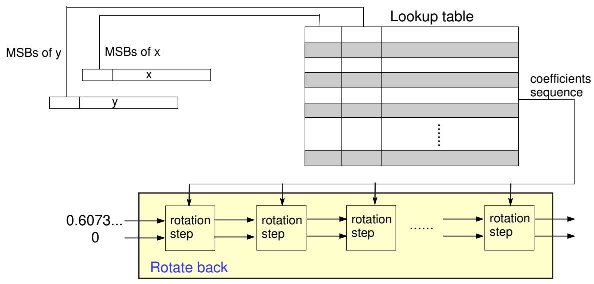
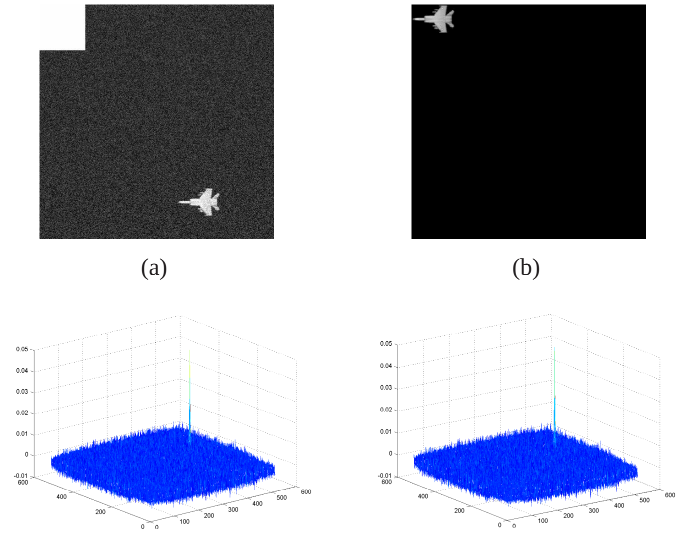


Fig. 4. A hybrid solution of lookup table and CORDIC

C. Reduced Bit Number for Phase Representation

The operation of taking MSBs of x and y actually rounds the phase to a less accurate digital representation,



(a) (b)

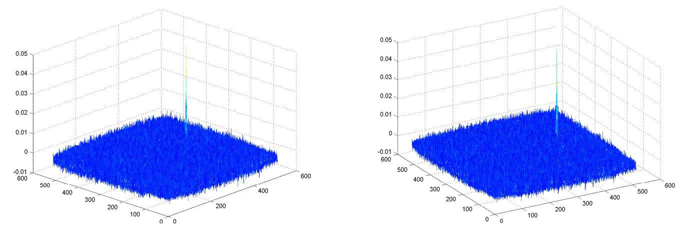
(c) (d)

Fig. 5. Comparison of CORDIC and hybrid solutions

though the CORDIC rotation is accurate. That means that the CORDIC rotation can only rotate the vector to the rounded the phase. In that case, there is no point to have such an accurate operation after an inaccurate one. Therefore, we can totally remove the CORDIC from the algorithm and use only the lookup table. The index of the table can still be the MSBs of x and y while the contents of the table should be modified to normalized x and y with 24 bits precision.

D. Sign Bit Only (SBO) solution

In the previous section, the experimental results show that using 5 MSBs to represent x and y works well. In fact, even using fewer bits, the result can still be good enough to detect the peak. Figure 6 gives some examples of SPOMF using fewer bits representation. Figure 6(a) shows the correlated image of the mig-25 fighter in 4 bits phase representation and figure 6(b) shows the 3 bits phase.



(a) (b)

Fig. 6. SPOMF with phase representation of fewer bit number

Since using 4 bits or even 3 bits can still provide very good results, it is natural to try the extreme case: only 1 bit. In that case, only the sign bit of x and y are taken into consideration. For instance, if the sign bit of x is 1 (negative) and the sign bit of y is 0 (positive), the output angle is 135 degree, meaning that the normalized complex number is $-\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}i$. In fact, there are totally four possible output numbers of the normalization operation, they are $\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}i$, $-\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}i$, $-\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}i$ and $\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}i$ whatever the input is.

In such extreme case, the correlated image after SPOMF still gives almost the same quality as high accuracy representation or even as the full CORDIC. MatLab simulation shows that the magnitude of the peak reduced by 9.96% compared to full CORDIC for different images. Figure 7 gives the same example of the mig-25 fighter image using sign bit only (SBO) normalization. Figure 8 gives another example of a more complex image. The search image contains a lot of apples and the template image has an apple in the top left corner. Notice that no scale and rotation compensation is used here but only the regular SPOMF. Figure 8(c) is the correlated image of full CORDIC normalization while 8(d) uses SBO normalization.

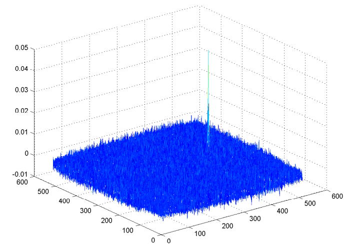
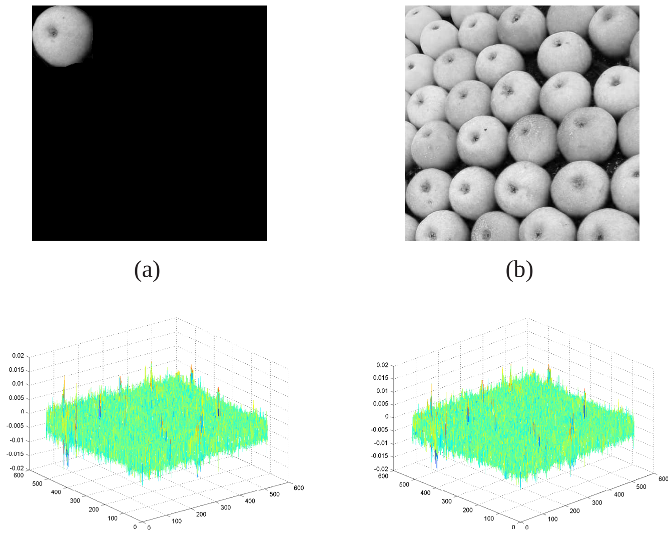


Fig. 7. SPOMF of mig-25 using sign bit only (SBO) normalization



(a) (b)

(c) (d)

Fig. 8. Comparison of SPOMFs with full CORDIC and SBO

E. Mathematical Explanation

To explain why the SBO works so well, we have to understand the actual mechanism of how phase difference represents the shift information of two images.

After Fourier transform, the search image and the template image in the frequency domain will be conjugate multiplied. In the result, that is a 2 dimensional matrix with complex numbers. The horizontal shift is represented by the number of circles that the phase rotates and the vertical shift, the same as horizontal shift, is represented by the number of circles that the phase rotates along the vertical direction. Figure 9 gives an example of the phase of such a matrix when there is a shift of 5 pixels along the horizontal direction and 9 pixels along the vertical direction. With multiple objects in the search image, the phases that represent multiple translations will be added up. For example, if there are objects located in position (5, 9) and (8, 12), the phase in the horizontal direction will be the sum of a phase function with 5 cycles and another phase function with 8 cycles.

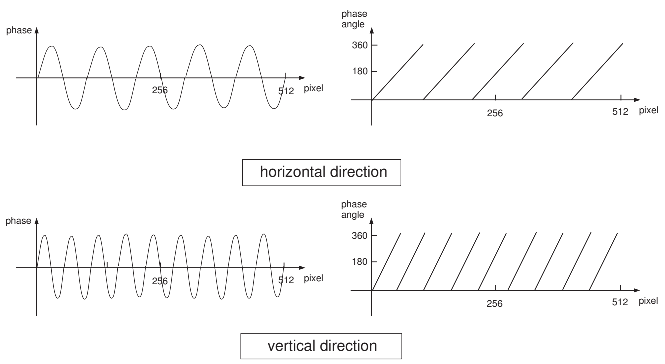


Fig. 9. Shift information represented by phase rotation circles

Now, let's use the SBO method to quantize the phase angles to 4 possible angles (45, 135, 225 and 315 degree). From figure 10, we can see that although some individual phases lose large amount of information from their original values, the overall characteristic of all the phases is still kept well. The total number of circles that the phase has rotated can barely change from the quantization.

The SBO normalization can also be mathematically proved to be reliable. Let $F(\xi, \eta)$ be the frequency domain representation of the template images. Then the search image which differs by a displacement of (x_0, y_0) can be represented by $F(\xi, \eta) \bullet e^{-j2\pi(\xi x_0 + \eta y_0)}$. After conjugated multiplication, the correlated matrix is like:

$$F(\xi, \eta) \bullet \overline{F(\xi, \eta)} \bullet e^{-j2\pi(\xi x_0 + \eta y_0)} \Longrightarrow |F(\xi, \eta)|^2 \bullet e^{-j2\pi(\xi x_0 + \eta y_0)}$$

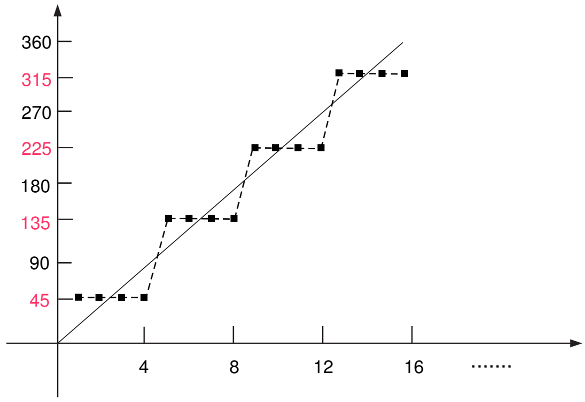


Fig. 10. The well kept overall characteristic with SBO normalization

The regular normalization method normalizes the $|F(\xi, \eta)|^2$ factor to 1, and the correlated matrix becomes only $e^{-j2\pi(\xi x_0 + \eta y_0)}$. Let $Err(\xi, \eta)$ be the error function in the frequency domain caused by the phase quantization. Then the correlated matrix becomes $e^{-j2\pi(\xi x_0 + \eta y_0) + Err(\xi, \eta)}$ after SBO normalization. It can be converted to $e^{-j2\pi(\xi x_0 + \eta y_0)} \bullet e^{Err(\xi, \eta)}$. Note that a multiplication in the frequency domain is a convolution in the spatial domain. So if we take the inverse FFT to the correlated matrix, it becomes:

$$IFFT(e^{-j2\pi(\xi x_0 + \eta y_0)}) * IFFT(e^{Err(\xi, \eta)})$$

The left term in the above correlated image (note that we use correlated matrix for the frequency domain and correlated image for the spatial domain) is the correct peak. So let us see what happens when the correct peak is convoluted to the right term. When we quantize the phase to 4 angles using SBO normalization, the distortion is no more than $\pi/4$. If the errors are randomly distributed within the range of $-\pi/4$ to $\pi/4$, the inverse FFT of this error function is a strong DC component plus some low level noise. The convolution of the correct peak and the DC component still gives a peak in the correct position. Since the magnitude of the DC component is slightly less than 1, the magnitude of the convoluted peak will also be reduced. The more bits we use to represent the phase, the higher the convoluted peak is. Table III-E gives a comparison of different peaks' magnitude under different bit number representations.

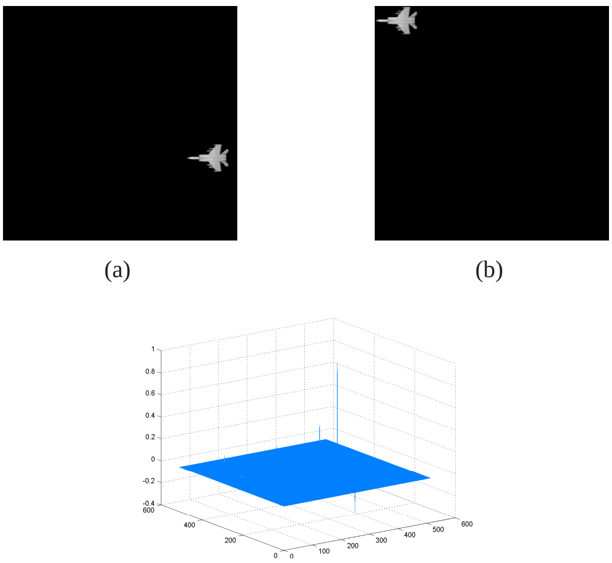
In the above description, we assume that the error function $Err(\xi, \eta)$ is a random distribution in the range of $-\pi/4$ to $\pi/4$. For most of the applications, this model is quite close to the real error distribution because either the the noise in the image or the background information can cause the phase angle fluctuates along the linear model in

	SBO	2 bits	3 bits	4 bits	5 bits
Peak percentages	90.04%	97.45%	99.36%	99.84%	99.96%
Noise level increments	0.076%	0.036%	0.019%	0.010%	0.005%

TABLE II

PEAKS' MAGNITUDE AND NOISE LEVELS UNDER DIFFERENT BIT NUMBER REPRESENTATIONS

the right part of figure 9. But in some cases when the background is very clean and the noise level is very low, meaning a very nice match, the quantization of the phase angle will be very regular like in figure 10. This stairs-shape regularity in the frequency domain is actually a regular pulse which can cause some peaks other than the correct one in the spatial domain after inverse FFT. See figure 11. Both the search image and the template image have clean background. We can see that the correlated image has some faulty small peaks which actually indicate the regular pulse in the frequency domain.



(a) (b)

(c)

Fig. 11. Faulty peaks in the clean match

Such a nice match does not happen often in practice. The log-polar mapping, the compensating rotation and scale all introduce interpolation errors. Experiments show that even small errors can remove this regular pulse. If there is no rotation and scale in the search image, background information will also introduce irregularity, which removes the faulty peaks eventually. The only chance, which seldom happens, is that both the search image and the template image have a clean background without rotation and scale. But still, the peak detection algorithm can remove those faulty peaks in a later stage [1]. It is safe to say that the SBO normalization is a reliable algorithm for this application.

IV. FUTURE WORK

From the theory section we can see that the method we use is simply using the phase to detect the translation and using the magnitude to detect rotation and scale. In fact, rotation and scale information is present both in the phase and the magnitude. Therefore, better information abstraction techniques can be developed to greatly enhance the performance of this algorithm. The whole algorithm can be extended to 3 dimensional spaces with the same techniques (using a 3D camera). The only problem is that it requires a huge amount of calculations. We can hope that future powerful chips and systems will be available to perform the tasks.

REFERENCES

[1] Meng Ma, Arjan van Genderen and Peter Beukelman, *Developing and Implementing Peak Detection for Real-Time Image Registration*, ProRISC 2005

[2] Peter Beukelman and Laurens Bierens, *Real-Time Rotation and Scale Invariant Template Matching in Streaming Images*, GSPx (2004)

[3] B. Srinivasa Reddy and B. N. Chatterji, *An FFT-Based Technique for Translation, Rotaion, and Scale-Invariant Image Registration*, IEEE Transaction on Image Processing, Vol 5, No. 8, page 1266-1271 August 1996.

[4] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and J. Waldn *Polar FFT, rectopolar FFT, and applications*, Stanford Univ., Stanford, CA, Tech. Rep., 2000.

[5] Javier Valls, *Evaluation of CORDIC Algorithms for FPGA Design*, Journal of VLSI signal processing 32, page 207-222, 2002.

[6] Israel Koren, *Computer Arithmetic Algorithms, 2nd Edition*, page 233-234, 2002.

[7] Morgen McGuire, *An Image Registration Technique for Recovering Rotation, Scale and Translation Parameters*, Massachusetts Institute of Technology, Cambridge, MA, 1998.