2020「中技社科技獎學金」
2020 CTCI Foundation Science and Technology Scholarship
境外生生活助學金
Living Grant for International Graduate Students

RST Template Matching-Based PCB Alignment System on Embedded GPU Board

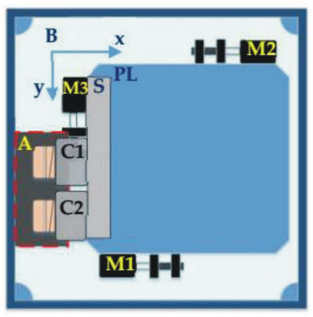
Minh-Tri Le, and Jenn-Jier James Lien

Department of Computer Science and Information Engineering, National Cheng Kung University

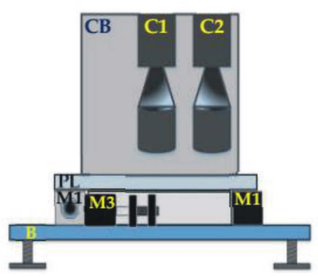
Abstract

The present study proposed an embedded printed circuit board (PCB) alignment system, in which a rotation, scale and translation (RST) template-matching algorithm was employed to locate the marks on the PCB surface. The coordinates and angles of the detected marks were then compared with the reference values which were set by users, and the difference between them was used to adjust the position of the vision system accordingly. To improve the positioning accuracy, the angle and location matching process was performed in refinement processes. To overcome the matching time, in the present study we accelerated the rotation matching by eliminating the weak features in the scanning process and converting the normalized cross correlation (NCC) formula to a sum of products. Moreover, the scanning time was reduced by implementing the entire RST process in parallel on threads of a graphics processing unit (GPU) by applying hash functions to find refined positions in the refinement matching process. The experimental results showed that the resulting matching time was around 32× faster than that achieved on a conventional central processing unit (CPU) for a test image size of 1280 × 960 pixels.

Overview System

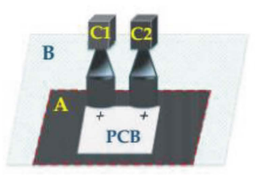


Top view of the vision alignment platform

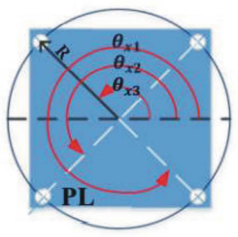


Front view of the vision alignment platform

- ***B***: system base
- ***M1, M2, M3***: three stepper motors
- ***C1, C2***: two cameras
- ***PL***: movable platform
- CB: camera base
- ***A***: location of PCBs
- ***x, y***: axis movements of the movable platform
- ***R***: the radius from the center of the platform to the center of the motor axis
- θ_{x1}, θ_{x2}, and θ_{y}: the angles between the axis of motors M1, M2 and M3, respectively, relative to the x-axis of the platform



Location of PCB in the system



Description of parameters of the movable platform

NVidia Jetson TX2 development kit

Figure 1. Overview of the hardware of the proposed embedded printed circuit board (PCB) alignment system.

RST Refinement Template Matching Algorithm

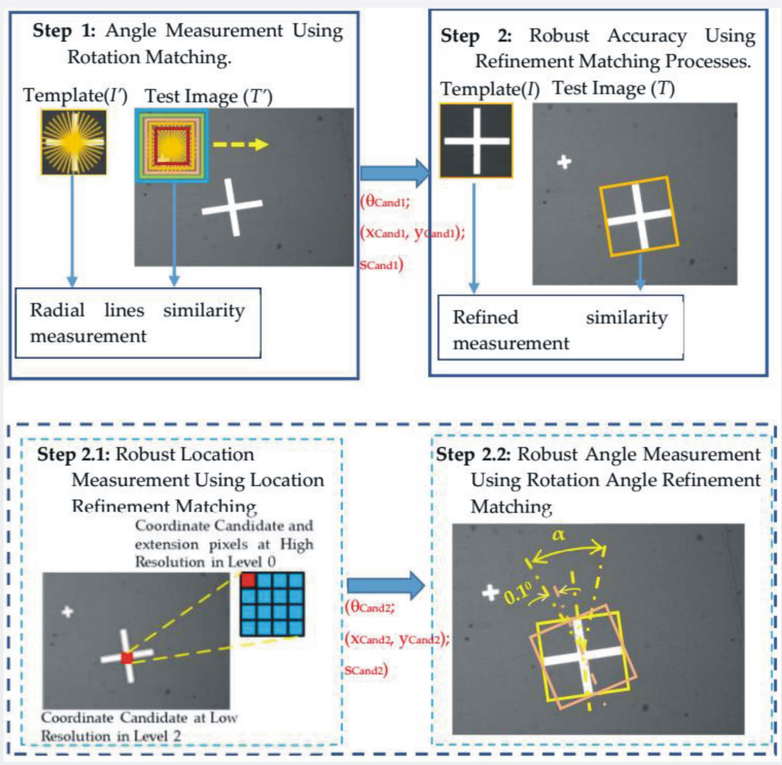


Figure 2. Global framework of the RST refinement template matching algorithm.

Acceleration of the RST Refinement Template Matching Algorithm

Algorithm 1 Pseudo-code of Location Refinement Matching using CUDA

Inputs: Test image I, size of template $w \times h$, coordinate candidates (x_{Cand1}, y_{Cand1}), angle candidates θ_{Cand1}, scale candidates s_{Cand1}, pyramid level N_P, number of candidate N_{Cand1}
Outputs: Correlation coefficient η_{NCC}
X index: $idxX \leftarrow blockDim.x * blockIdx.x + threadIdx.x$ //Number of extension pixels
Y index: $idxY \leftarrow blockDim.y * blockIdx.y + threadIdx.y$ //Number of candidates
if ($idxX < 2^{N_P}$) *and* ($idxY < N_{Cand1}$) **then**
  Coordinate X: $x_{Refined} \leftarrow (idxX \bmod 2^{N_P}) + (x_{Cand1} + 2^{N_P})$ //Refinement coordinate x
  Coordinate Y: $y_{Refined} \leftarrow (idxY \operatorname{div} 2^{N_P}) + (y_{Cand1} + 2^{N_P})$ //Refinement coordinate y
  Scale s: $s \leftarrow s_{Cand1}[idxY]$
  Angle θ: $\theta \leftarrow \theta_{Cand1}[idxY]$
  for j in h **do**
    for i in w **do**
      Collect intensity pixel values inside a search window with a center point at ($x_{Refined}$, $y_{Refined}$), an orientation: θ, and a scale: s
    end
  end
  Calculate the NCC score η_{NCC} between the template and the search window
end
Return η_{NCC}

Algorithm 2 Pseudo-code of Rotation Angle Refinement Matching using CUDA

Inputs: Test image I, size of template $w \times h$, coordinate candidates (x_{Cand2}, y_{Cand2}), angle candidates θ_{Cand2}, scale candidates s_{Cand2}, angular resolution α, number of candidate N_{Cand2}
Outputs: Correlation coefficient η_{NCC}
X index: $idxX \leftarrow blockDim.x * blockIdx.x + threadIdx.x$ //Angular resolution
Y index: $idxY \leftarrow blockDim.y * blockIdx.y + threadIdx.y$ //Number of candidates
if ($idxX < \alpha * 10$) *and* ($idxY < N_{Cand2}$) **then**
  Angle θ: $\theta_{Refined} \leftarrow (idxX \ / \ 10.0) + \theta_{Cand2}[idxY]$ //Refined angle
  Coordinate X: $x \leftarrow x_{Cand2}[idxY]$
  Coordinate Y: $y \leftarrow y_{Cand2}[idxY]$
  Scale s: $s \leftarrow s_{Cand2}[idxY]$
  for j in h **do**
    for i in w **do**
      Collect intensity pixel values inside a search window using a bilinear interpolation approach with a center point at (x, y), an orientation: $\theta_{Refined}$, and a scale: s
    end
  end
  Calculate the NCC score η_{NCC} between the template and the search window
end
Return η_{NCC}

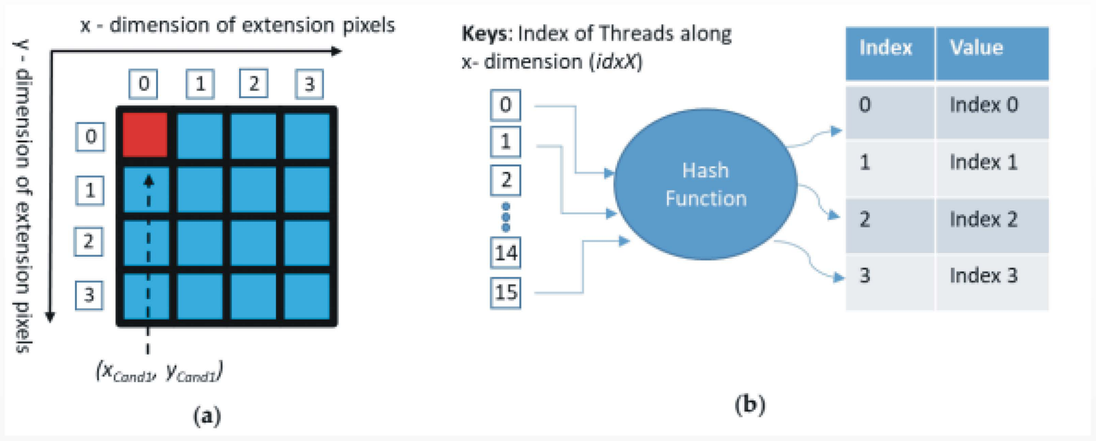


Figure 3. Reducing matching time by using the hash table for CUDA coding. (a) Description of coordinate candidate (x_{Cand1}, y_{Cand1}) and extension pixels; (b) description of finding x and y indexes of the extension pixels.

Experimental Results

Table 1. Comparison of average matching time and accuracy of RST template matching on different platforms and with different image sizes.

Methods	Image Size					
	640x480 (pixels)		800x600 (pixels)		1280x960 (pixels)	
	Time(s)	Accuracy	Time(s)	Accuracy	Time(s)	Accuracy
PC-based Platform						
PC-RST[1]	0.618	98.0%	1.075	98.0%	1.326	97.5%
FAsT-Match[2]	0.1	99.9%	-	-	0.4	99.8%
PC- Improved RST	**0.099**	97.5%	**0.186**	95.5%	**0.291**	97.0%
Embedded System-based Platform						
em-RST	1.914	97.0%	5.668	96.5%	9.517	95.0%
emCPU-Improved RST	0.568	92.0%	1.633	95.0%	3.664	98.0%
emGPU-Improved RST	**0.197**	96.5%	**0.342**	95.5%	**0.301**	96.0%

The results show that the performance of the proposed method is faster than others methods on both PC platform (*PC-Improved RST*) and embedded platform (*emGPU-Improved RST*), while having no significant effect on the accuracy of the matching results.

References

[1] Le, M.T.; Li, C.H.; Guo, S.M.; Lien, J.J. *Embedded-Based Object Matching and Robot Arm Control*. In Proceedings of IEEE Int. Conf. Auto. Sci. Engi. (CASE), 2019; pp. 1296-1301.

[2] Liu, B.; Shu, X.; Wu, X. *Fast screening algorithm for rotation invariant template matching*. In Proceedings of 25th IEEE Int. Conf. Image Processing (ICIP), 2018; pp. 3708-3712.

Acknowledgement

This research was funded by the Ministry of Science and Technology (MOST), Taiwan, R.O.C., by Contrel Technology Co., Ltd (Taiwan) and Tongtai Machine & Tool Co., Ltd (Taiwan).