10.2352/EI.2022.34.17.3DIA-223
© 2022, Society for Imaging Science and Technology

Scale-Adaptive Local Intentional Surface Feature Detection*

Yujian Xu, Robert Ulichney, and Jan Allebach; Purdue University, ECE Department; West Lafayette, IN
Matthew Gaubatz; StockX; Seattle, WA
Stephen Pollard; HP Labs; Bristol, UK

Abstract
Adding features to the surface of a part creates opportunities to serialize the part with an identifier, and/or to provide enhanced measurements of the surface geometry, provided that the features can be detected. Scale-space filtering is a common tool used with both pixel and voxel representations of objects for such purposes. One challenge associated with extending this class of successful algorithms to detect features on images of 3D surfaces in 2D imagery is that the filtering is performed in the plane of the camera sensor, which may not be conveniently related to the orientation of the target object's surface. If the object is represented with 3D voxel data, creating a similar effect can also require a similar relationship between the plane of the local surface and the orientations of the voxels relative to the target object. Thus, a mesh-based approach is desirable. Furthermore, whereas many feature detection schemes target scale-invariant features, a desired outcome is the ability to localize features intentionally created with deformations at a given scale, that is, scale-specific artifacts. This paper proposes a technique for adding features of a known size to a 3D mesh representing an object, then adopting the ability to compute per-scale differences to the local mesh surface geometry to match a known feature of a known scale. We introduce a tunable two-scale depth measurement scheme to quantify the displacement of a vertex from the local surface, which can be a strong indicator of features. We print and scan 3D models with fiducial features appearing across the surface to demonstrate the fidelity and accuracy of the proposed feature detection scheme.

Introduction
Technologies for 3D printing parts are becoming increasingly common in both consumer and industrial applications. When parts are designed, modeled, printed, and inspected (scanned), 3D representations of the parts, such as voxels and meshes, serve many useful purposes. In particular, the ability to detect features from these 3D representations can enhance many tasks such as surface matching [1], shape retrieval, viewpoint selection [2], motion capture, animation synthesis, and registration of the scanned mesh with a CAD design.

While some feature detection algorithms are highly performant on 2D data, such as the Harris corner detector [3] and the scale-invariant feature detector [4], creating extensions to our 3D data is not trivial. It is straightforward to apply filtering operations if the parts are represented using voxels, but in some cases quality voxelation requires uniform sampling. For some types of mesh designs, this requirement may not be satisfied. First, the length of the edges in the mesh can be arbitrary. Second, the local topology in 3D meshes is relatively unconstrained, that is, a vertex can have an arbitrary number of neighboring vertices. While it is possible to convert meshes to voxels and vice versa, this conversion may result in a loss of precision. Still, this approach has been applied such as in [5], where an extension of the SIFT detector to voxels is proposed. The first step in their scheme is to voxelize the input mesh. While filtering voxelized meshes can show promising results, it can also present computational as well as performance drawbacks. High resolution voxelization is expensive in terms of both time and memory, and furthermore makes the subsequent processes, such as scale-space filtering, less efficient. In addition, the response of scale-space filtering in voxels can be weak, that is, resulting in spurious detection and missing some key points.

As mentioned above, the main challenges of mesh-based detection approaches are due to the irregularities of the mesh, some of which can be addressed in a variety of ways. One is to approximate the the surface using local imagery around a given vertex. In [6], for instance, ring structures are defined in the mesh to find the neighborhood of a vertex around which a local plane is fit. After translation and rotation, the centroid of the neighborhood is set to the origin and the normal of the plane is along the z-axis. Then, the local mesh can be interpreted as a depth image over this local (XY-)plane. Other approaches involve creating scale-space representations of the mesh via decimation [7] and smoothing [1, 8, 9]. In [1], the smoothing is done directly in the geodesic scale space without resorting to surface mapping or parameterization procedures to handle the irregularity of the mesh.

Most of the above works focus on detecting existing features of the 3D representations, while few focus on the importance and use-case of detecting intentionally created features. Intentional features are small structures of potentially any shape that are added to the surface of a part. For example, intentional features can be arranged on the surface to carry some information, or they can be randomly spread for enhancing the further measurements of the surface geometry. Intentional features do form a special case of surface features, but a practical challenge associated with using them is that they often must not interfere with the intended functionality of the part. Furthermore, they may appear at the same scale as existing part features. Hence, general feature detection algorithms can detect the intentional features to some degree, but may result in localization degradation and spurious detections compared to an algorithm designed specifically for this purpose.

In this paper, we present a scheme designed to localize intentional features at a given scale. A key property of this approach is the notion of local planarity, as discussed in [6], where local 3D meshes are reduced to 2D imagery with little computation compared to the processing applied in the geodesic space. In our own approach, instead of addressing surface mapping by translation and rotation of the vertices in the local patch, we calculate a depth

*Research supported by HP Inc., Palo Alto CA. U.S.A.

value directly for each vertex by selecting two patches around the vertex representing different scales. The sizes of the patches are related to the given scale of the intentional features. Depth values assigned to each vertex form a scalar function over the mesh, and the structure of this function serves as a strong indicator of the intentional features; the features can be detected as local extrema that satisfy constraints applied by several additional filters.

This paper is organized as follows. Section 2 summarizes previous work involving intentional features. Section 3 presents a description of our detection method. Section 4 presents the experimental results, and Section 5 concludes the paper.

Intentional Features

In 3D objects, intentional features often serves as fiducial markers. A common example includes use on PCBs [13, 14]. While the question of what constitutes a robust design for fiducials to be detected from 2D images has been studied [15], our problem differs in that the printing process is truly monotone, and the feature must be created as a deformation of a surface. Given a position on the part, we can deform the local patch around it to create an intentional feature. The shape and size of the deformation can be arbitrary, but we have two goals: to minimize the distortion of the geometry of the part, and to maximize the number of the intentional features that can be detected after printing. The first goal requires regular and symmetrical shape and a size as small as possible. The second requires the exact opposite characteristics; and it depends on the printing device and detection algorithm. It is difficult to evaluate the two goals analytically, so we decided to print different intentional features on planks to select a reasonable shape and size. We selected two shapes: semi-sphere and chisel. The size of the feature varies from 0.5mm to 2mm. The planks were printed by the HP 3D fusion printer. Figure 1 presents the CAD mesh of an example plank with different intentional structures. Visual inspection of the test sets suggest the semi-sphere features provides better printing quality and are more perceivable than the chisel feature at the same scale. The semi-sphere features of 1mm diameter are clearly printed. The features smaller than these may be degraded after printing and may be hardly perceivable.

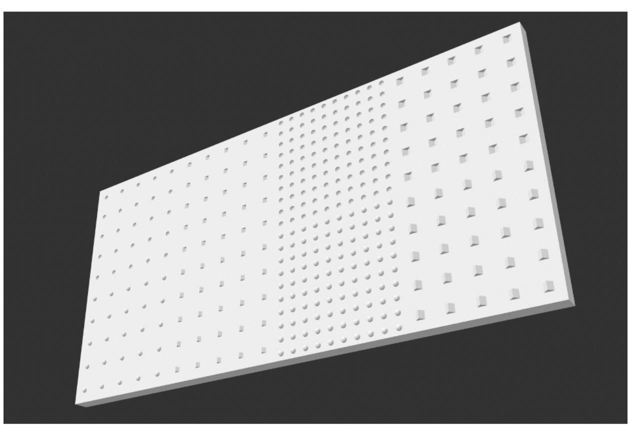

Figure 1. *An example plank with different intentional structures.*

One application of intentional features are for encoding data. These have been used to mark circuit components [16], and in a recent application, have been applied as physical scene markers to augment LiDAR measurements [17]. Our previous work [10] proposed a scheme that uses a 2D grid-like point pattern to encode a message, as is shown in Figure 2, which we have extended for rendering onto 3D surfaces [11]. The encoding scheme determines the surface positions at which the deformations are placed. Figure 3 shows a sphere, a cylinder, and a tile with a quasi-random surface. The intentional feature patterns encoded with messages are added to their surfaces. The grid-like patterns are distorted by the curved surfaces after mapping in the imagery, which presents a significant challenge to the image-based decoding process. In particular, it is necessary to detect these intentional features in the imagery, i.e., a 2D feature localization algorithm is needed. We adapted the SIFT algorithm for the intentional features to serve this purpose; for details relating to data encoding and feature alignment see [11]. The process of creating, rendering and later on interpreting these intentional features is referred to as *surface coding*.

Figure 2. *An example of a portion of a 2D data-bearing point pattern.*

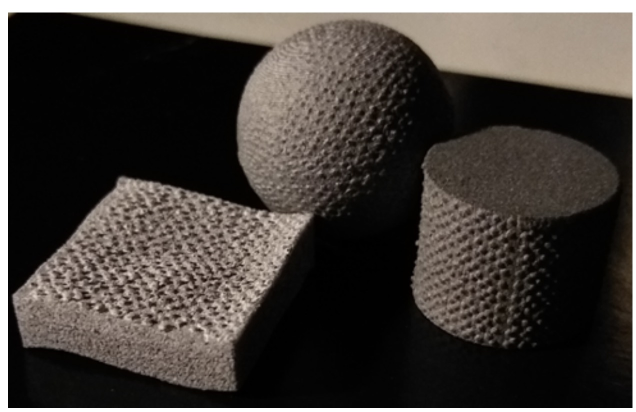

Figure 3. *3D printed parts with surface coding*

Another application of the intentional features is to enhance the measurement and analysis of the geometry of a targeted region

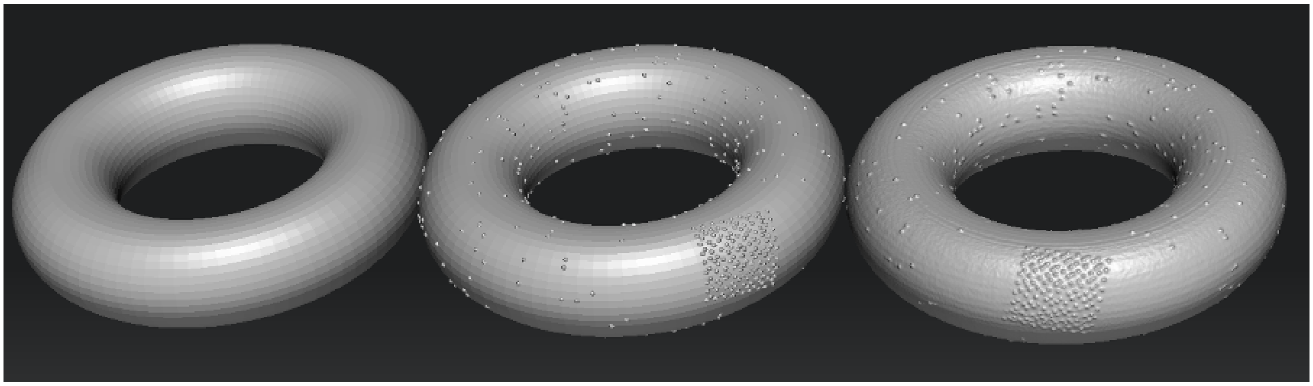

Figure 4. Original (left), modified (center) and scanned (right) meshes. The features in the scanned mesh are less cleanly rendered than in the digital original, but are still easily detectable with the proposed approach.

of a manufactured part [18], or in some cases placement within a scene [17]. While these applications involved fiducials characterized by variations in signal intensity, focus herein is on modifications of (monotone) surface designs. We can, for example, add intentional features to the surface of a part, especially if the part lacks enough native descriptive features to help register surface measurements. An extreme example involves toroidal parts, also referred to as donuts. No native features can be detected on the ideal torus due to the symmetry, i.e., it is possible to characterize the surface, but it is difficult to tie such measurements to any local (absolute) locations on the surface. Adding intentional features creates asymmetries that help mitigate this issue, not only with respect local geometric measurement, but also for registration between a 3D CAD design and a scan of the part. In Figure 4, the original, modified, and scanned meshes of a torus are shown. Some intentional features are added at the random positions on the surface, while others represent a part ID via surface coding. Interpreting these features correctly hinges on the ability to accurately detect and localize them.

Detection Method

The feature localization scheme is presented in the context of a system that creates parts with intentionally determined scales. A key aspect is that the feature size and specifications (geometry, direction with respect to the surrounding surface, etc.) are controllable. Given a design for a part and the desired feature specifications, a modified part is created with surface augmentations at a particular scale. This use case is representative of arrangements in which the designer of a 3D part specifies the overall geometry of a part, but admits a degree of flexibility for including labels and/or other adjustments to assist production. Once that part is printed and scanned, the feature detection scheme is used to estimate the same feature locations from measurements of the physical rendering of the part. Note that the detection scheme can also be applied to a modified mesh.

The detection method is described using a graph that is used to model a part mesh. Given a mesh M containing a set of vertices V and a set of faces F, we can model this part with a graph G, as each node represents a vertex and each face constructs three edges in G. For subsequent processing, it is helpful to consider the separation between nodes in terms of both graph characteristics and distance in three-dimensional space. Accordingly, the edge distance between two nodes, denoted by $d_E(v_1,v_2)$, is the minimal number of edges between them. The Euclidean distance between two nodes in 3D space is denoted by $d_2(v_1,v_2)$. A local region of the mesh around a vertex v can be defined as all the vertices within some distances of a given distance of v, such as a neighborhood of all points v' such that $d_E(v,v') \leq N$, where N is a non-negative integer. These vertices are denoted by a set $S(v,N)$. An example is shown below in Figure 5.

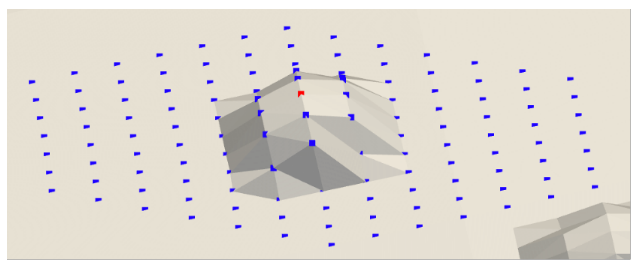

Figure 5. The vertex v is shown as the red point. A set $S(v,7)$ is shown as the blue points and red point together.

To find the intentional features on the smooth surface, we assign a depth value $f(v)$ to each vertex. The depth value quantifies the displacement of a vertex from the local surface. To be more specific, it is the perpendicular distance from a vertex to the local best fit plane. For a vertex v, we find its first neighborhood as a set of vertices $S(v,N_H)$. The value of N_H is pre-determined by the scale of the intentional features. Then we find all the mesh faces $\{F_i\}$ associated with the vertices $S(v,N_H-1)$. Let the centroid of each face be given by $C(F_i)$ and the area of each face be denoted by $A(F_i)$. We calculate the centroid of the neighborhood region using

$$c_H = \frac{\sum_i A(F_i)C(F_i)}{\sum_i A(F_i)}, \qquad (1)$$

an area-weighted average of the per-face centroids of all faces within the region.

Next, we find the nearby vertices in a smaller region $S(v,N_L)$ than the first one, i.e., for which $N_L < N_H$, and calculate an area-weighted centroid c_L via the same approach. In our experiments, simply setting $N_L = 1$ provides good performance. The depth

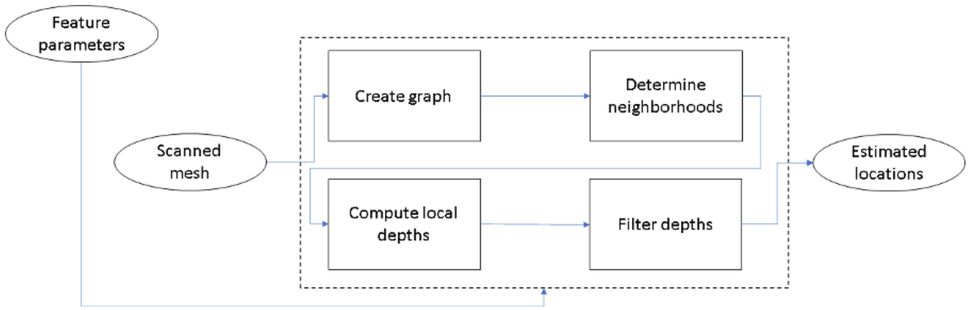

Figure 6. *System diagram of the feature localization process. Note that structures such as KD-trees can be used to compute estimated locations with improved efficiency.*

value of v is given by

$$f(v) = \mathbf{n}^T(c_H - c_L)w(v), \tag{2}$$

where $w(v)$ denotes a normalization weight selected based on the mechanism for establishing a local neighborhood. If the neighborhoods are established based on edge-distances, for instance, $w(v)$ is set to be inversely proportional to the square root of the total sampled area to handle the non-uniform meshes, i.e.,

$$w(v) = \frac{1}{\sqrt{\sum_i A(F_i)}}, \tag{3}$$

Without this normalization, the local depth value at smoothly curved surface regions without features could be rated as higher than that at a feature position. For an approximately uniform mesh, the normalization can be simplified to $w(v) = 1$. In such a mesh, the above method will compute an effective bandpass filter over the surface depth. If the mesh does not satisfy this property, however, other means must be used to control the (physical) span of the sets $S(v,N_H)$. In particular, one can enforce the constraint that $u \in S(v,N_H)$ implies that the distance $d_2(v,u)$ is bounded above by some threshold. A convenient practical threshold is calculated as twice the radius of intentional features.

Once a set of feature depths has been generated, we post-process the candidate vertices using three tests. First, we pre-filter the candidates according to their depth values with a constant threshold T_{abs}, i.e., by only retaining vertices v such that $f(v) \geq T_{abs}$. The threshold can be determined by the size of the features as designed. Second, among the remaining vertices, we find and retain the local extrema of $f(v)$. That is, for each candidate feature position v, we find all other candidate feature positions v' in $S(v,N_1)$ for some N_1, and if

$$\frac{f(v)}{max\{f(v')\}} \leq 0.5, \tag{4}$$

this candidate is discarded. Finally, the candidate features that are too close to each other are merged to be a single position, that is, if candidate feature positions v_1 and v_2 satisfy $d_E(v_1,v_2) \leq T_{same}$, they are effectively assigned to the same feature cluster. For candidate feature positions within each cluster, we calculate their averaged position as the final detected feature position.

A system-level diagram summarizing this approach is given in Figure 6. A key step in the estimation scheme involves breaking up the entire mesh surface into local neighbourhoods, as described above. Graph structures yield one convenience in doing so, i.e., the ability to disambiguate different faces of object components that are near each other. As a result, we will not accidentally use vertices associated with nearby neighboring structures (such as, for instance, adjacent fins on a CPU cooling device) to determine local planes.

Experimental Results

The whole experiment follows the three steps below:

1 We created 3D part models with surface fiducial structures; locations were stored as the ground truth.
2 Fiducial structure positions were detected from the CAD mesh and compared with the ground truth.
3 The models were 3D printed, scanned, and compared again to the ground truth.

The first two steps are trivial while the last step needs extra alignment. After printing and scanning, the whole model may be displaced, rotated and distorted. The non-linear distortion can hardly be recovered; but we can estimate the translation and rotation, and then build the correspondence between the detection result and the ground truth. We investigated two methods. First, we directly applied the non-rigid CPD [12] method to register the detected positions of intentional features with the ground truth positions. However, the CPD method cannot reliably estimate the transformation for parts with symmetrical shapes. For example, the CPD iteration terminated too early for the torus part even if we used strict termination conditions. We assume the optimization is trapped in the local minimum surrounded by a plateau.

One alternative is to extend our grid alignment algorithm [10] to the 3D case and even the non-grid case. The original algorithm can only align a set of 2D points with a grid structure existing in it. While our target point set is in the 3D space, and no grid structures are guaranteed to exist. We successfully extended the grid alignment algorithm to 3D, but a grid structure was still required. For the convenience of the experiments, we selected a patch on each part that is used for surface coding. In Figure 4, a square patch can be clearly seen. After we detected the intentional features, we first aligned the patch with the 3D grid alignment algorithm. Then, translation and rotation could be estimated based on the aligned points compared to the stored ground truth positions in the patch. Finally, we transformed the whole set of ground truth positions with the estimated translation and rotation.

CAD meshes

For the sake of saving storage, the CAD meshes are usually non-uniform. The sub-mesh of the smooth region or region with a small curvature has a lower resolution than others. The irregularity may hamper the detection. However, the CAD meshes are ideal representations of the parts. The intentional features are not degraded by printing and scanning.

We compared our approach with LDSIFT [8], where a similar local depth concept is proposed, on the CAD meshes of the torus. The result of our method was nearly perfect. All the intentional features are detected and the detected positions are generally located at the center of each feature, as shown in Figure 7. The LDSIFT missed only a few intentional features, but it also detected several spurious features. Moreover, the positioning of each detection was also worse.

Figure 7. *Close-up view of detected features using the proposed approach (green) vs. LDSIFT [8] (red), which results in degraded localization and spurious detections.*

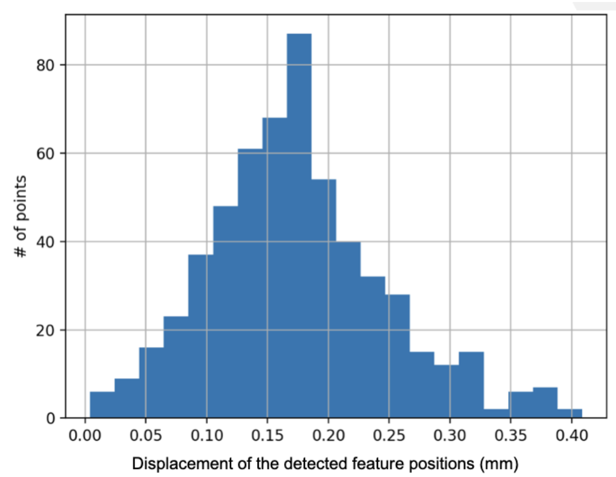

Figure 8. *Histogram of the displacement*

Real Prints

We printed and scanned some toroidal parts. The matching rate and the rate of missed detection were calculated based on the meshes derived from scanned parts. our method found 88% or more of the surface features, whereas LDSIFT matched fewer than 40%, and was either too aggressive in the wrong regions, or too conservative overall. We also calculated the displacement between the position of the true detection and its transformed ground truth position based on the alignment scheme mentioned above. A histogram of the displacement of one mesh is shown in Figure 8. The average displacement was 0.175mm as the radius of intentional structure is 0.4mm. Taking into account the distortion introduced by printing and scanning, the displacement purely caused by detection would be even less.

Since the feature detection performance is strong on non-flat regions of the surface, it yields an opportunity to embed data based on feature placement. Eight different messages were embedded in arrangements of features on toroidal surfaces using the coding method described in [10], and in each tested case, after printing, scanning and applying the proposed approach, the message embedded with the features was recoverable.

References

[1] G. Zou, J. Hua, M. Dong, H. Qin, Surface matching with salient keypoints in geodesic scale space, Computer Animation and Virtual Worlds 19(3–4), pg. 399–410 (2008).

[2] C. Ha Lee, A. Varshney, D. W. Jacobs, Mesh saliency, ACM Transactions on Graphics, pg. 659–666 (2005).

[3] C. Harris, M. Stephens, A Combined Corner and Edge Detector, Alvey Vision Conference. Vol. 15. (1988).

[4] D. Lowe, Object recognition from local scale-invariant features, Proc. of the Seventh IEEE International Conference on Computer Vision, volume 2, pg. 1150–1157 (1999).

[5] A. Godil, A. Wagan, Salient local 3D features for 3D shape retrieval, CoRR, volume 1105.2796 (2011).

[6] I. Sipiran, B. Bustos, Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes, Visual Computer, volume 27 (2011)

[7] D. Smeets, J. Keustermans, D. Vandermeulen, P. Suetens, meshSIFT: Local surface features for 3D face recognition under expression variations and partial data, Computer Vision and Image Understanding, Volume 117, Issue 2, pg. 158-169 (2013).

[8] T. Darom, Y. Keller, Scale Invariant Features for 3D Mesh Models, IEEE Transactions on Image Processing 21(5): 2758-2769 (2012).

[9] U. Castellani, M. Cristani, S. Fantoni, V. Murino, Sparse points matching by combining 3D mesh saliency with statistical descriptors, Comput. Graph. Forum 27(2), 643–652 (2008).

[10] Z. Zhao, Y. Xu, R. Ulichney, M. Gaubatz, S. Pollard, J. Allebach, Data-bearing halftone image alignment and assessment on 3D surface, In Color Imaging XXV: Displaying, Processing, Hardcopy, and Applications, (Part of IS&T Electronic Imaging) (2020).

[11] Y. Xu, M. Gaubatz, S. Pollard, R. Ulichney, J. Allebach, Atomic growing for grid alignment, NIP & Digital Fabrication Conference, Proc. of Printing for Fabrication, pg. 73-77(5) (2021).

[12] A. Myronenko, X. Song, Point Set Registration: Coherent Point Drift, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 32 (2010).

[13] S. Fonseka and J.A.K.S. Jayasinghe, Feature Extraction and Template Matching Algorithms Classification for PCB Fiducial Verification, volume 1 (2018).

[14] T. Le, C.-T. Tu 2, S.-M. Guo 1 and J. Lien, A PCB Alignment System Using RST Template Matching with CUDA on Embedded GPU Board, Sensors, volume 20 (2020).

[15] C. B. Owen, F. Xiao and P. Middlin, What is the best fiducial?, IEEE International Workshop on Augmented Reality Toolkit (2002).

[16] F. Digiacomo, F. Bologna, F. Inglese, C. Stefanini and M. Milazzo, MechaTag: A Mechanical Fiducial Marker and the Detection Algo-

rithm, Journal of Intelligent & Robotic Systems, volume 103, 2021.
[17] J.-K. Huang, J. Grizzle, Improvements to Target-Based 3D LiDAR to Camera Calibration, IEEE Access, volume 8 (2020).
[18] J. Mathur, S. Basu, J. Menold, and N. Meisel, Quality Assessment of Additively Manufactured Fiducial Markers to Support Augmented Reality-Based Part Inspection, ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering (2020).

Author Biography

Yujian Xu received his B.S. in mechanical engineering from Shanghai Jiaotong University. He is a Ph.D. candidate in the department of electrical and computer engineering at Purdue now. His work focuses on detection and alignment of 3D surface properties.