

## Local voxelizer: A shape descriptor for surface registration

Peng Song<sup>1</sup> (✉)

© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** Surface registration brings multiple scans into a common coordinate system by aligning their overlapping components. This can be achieved by finding a few pairs of matched points on different scans using local shape descriptors and employing the matches to compute transformations to produce the alignment. By defining a unique local reference frame (LRF) and attaching an LRF to shape descriptors, the transformation can be computed using only one match based on aligning the LRFs. This paper proposes a *local voxelizer* descriptor, and the key ideas are to define a unique LRF using the support around a basis point, to perform voxelization for the local shape within a cubical volume aligned with the LRF, and to concatenate local features extracted from each voxel to construct the descriptor. An automatic rigid registration approach is given based on the local voxelizer and an expanding strategy that merges descriptor representations of aligned scans. Experiments show that our registration approach allows the acquisition of 3D models of various objects, and that the local voxelizer is robust to mesh noise and varying mesh resolution, in comparison to two state-of-the-art shape descriptors.

**Keywords** shape descriptor; surface registration; scan alignment; 3D reconstruction

### 1 Introduction

With advances in depth sensing technology, it is now easy and flexible to acquire depth scans of real objects, e.g., using the Kinect [1, 2]. To acquire the 3D model, multiple scans of an object from different views must be captured in order to cover

the entire surface of the object. Surface registration is the process that brings these scans (i.e., partial surfaces) into a common coordinate system by aligning their overlapping components. However, scans could contain noise and overlap among scans could be small, making the problem challenging.

When the poses of scans are similar, the *iterative closest point algorithm* (ICP) [3, 4] can be used to register the scans. However, this usually requires manual effort to position the scans into coarse alignment or to accumulate multiple scans over time [1]. To handle input scans with arbitrary initial poses, many global registration methods [5–9] have been proposed. A common idea is to find a few matched points on a pair of scans by using geometric constraints or local features, and to employ the matched points to compute a rigid transformation to align the scans.

The point matching problem can be solved by using local shape descriptors, which are quantities computed for each point on the model surface based on the local shape around the point. Points with similar descriptors potentially correspond. Normally, a few (at least 3) corresponding points found by matching local shape descriptors are required to compute a rigid transformation. By defining a unique *local reference frame* (LRF) for each point using its support (i.e., the local shape around the point) and attaching the LRF to descriptors, just one pair of matched points can determine the transformation by aligning the three axes of their LRFs [10, 11]. This drastically reduces the search space for corresponding points (i.e., from at least 3 pairs to 1 pair), and thus increases the chance to find correct aligning transforms for the scans.

Taking advantage of a uniquely defined LRF [11], we propose a new shape descriptor, called the *local*

<sup>1</sup> School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China.

E-mail: songpeng@ustc.edu.cn (✉).

Manuscript received: 2015-08-17; accepted: 2015-08-29

*voxelizer*. The key idea is to first define a unique LRF using the support around a basis point and then perform voxelization for the local shape within a cubical volume aligned with the LRF. The descriptor is constructed by concatenating features extracted from local shape within each voxel. To find local features that ensure high descriptive ability of the descriptor, we propose a set of feature candidates, and quantitatively compare them. Thanks to the selected local feature and descriptor construction scheme, the local voxelizer is robust to mesh noise and varying mesh resolution, as validated by a quantitative comparison with two state-of-the-art shape descriptors [10, 12].

Based on local voxelizers, an automatic surface registration approach is proposed to acquire 3D models from a set of input scans. We first present a pairwise registration algorithm, which uses local voxelizers to represent scans and generate scan alignment candidates. Best alignment is selected from the candidates by estimating the degree of overlap of two transformed scans, which can be further refined using ICP [3]. A surface reconstruction method is further given based on the pairwise algorithm and an expanding strategy that merges descriptor representations of aligned scans. Experiments show that our registration method can acquire 3D models of objects with varying shape complexity.

## 2 Related work

Registration methods can be broadly classified as local registration approaches using ICP, global registration approaches that search for the best aligning transform, and registration based on local shape descriptors.

**Iterated closest points.** ICP is the classic local registration method which works by iteratively refining dense point correspondences [3, 4]. Many variants of ICP have been proposed to obtain more reliable correspondence in each iteration [13, 14]. Recently, Newcombe et al. [1] used ICP with GPU optimization for real time scanning. Bouaziz et al. [15] formulated a sparse ICP to robustly handle 3D data with outliers. However, ICP requires input scans that are coarsely aligned: otherwise, it may not converge to the correct solution.

**Global registration.** Surface registration can be formulated as a global problem to find the best aligning transforms for a set of scans. This is particularly useful when the scans are in arbitrary initial poses. An aligning transform can be uniquely determined by 3 pairs of (non-collinear) corresponding points. Therefore, one popular strategy is to employ RANSAC to find aligned triplets of point pairs [5] or 4-point congruent sets [2, 8]. Other methods express surface registration as an optimization problem with geometric constraints [6, 9, 16, 17].

**Local shape descriptors.** Local shape descriptors have a wide range of applications such as surface registration, shape retrieval, and 3D object recognition. These descriptors can be classified as low-dimensional and high-dimensional, according to the richness of the encoded local shape information.

*Low-dimensional descriptors.* Low-dimensional descriptors (i.e., purely local features), such as integral volume [7] and surface curvature [18], are easy to compute, store, and compare. However, their descriptive ability is limited since different points on the same scan can have very close descriptor values. Thus, a further disambiguating process is usually required for the potential correspondences built by these descriptors. Chua and Jarvis [19] used principal curvatures to assist the search for corresponding points on model surfaces. Gelfand et al. [7] used an integral volume descriptor to select feature points and compute their potential correspondences. Huang et al. [20] matched fractured object parts by using several integral invariant descriptors [21]. Albarelli et al. [22] used surface hashes for detecting feature points and aligning surfaces.

*High-dimensional descriptors.* High-dimensional descriptors provide a relatively detailed description of the shape around a surface point, and thus can be directly used to solve the correspondence problem. Johnson and Hebert [23] proposed a spin image representation by spinning a 2D image about the normal of a feature point and summing the number of points falling into each bins of that image. Huber and Hebert [24] further applied spin images to automatic surface registration. Frome et al. [25] proposed a 3D shape context descriptor based on accumulating 3D histograms of points within a partitioned sphere centered at a feature point. Mian

et al. [26] proposed a 3D tensor representation by constructing an LRF from a pair of oriented points and encoding the intersected surface area into a multidimensional table. More recently, Zhong [12] proposed intrinsic shape signatures by improving [25] based on a different partitioning of the 3D spherical volume and a new definition of LRF with ambiguity. Tombari et al. [10] proposed an SHOT descriptor by constructing a unique LRF for a feature point and concatenating local histograms defined on each bin within a 3D spherical volume aligned with the LRF. Guo et al. [11] constructed a local shape descriptor by rotationally projecting the neighboring points of a feature point onto 2D planes and calculating a set of statistics within a uniquely defined LRF.

Comparing with Refs. [10, 11] that also attach a unique LRF to shape descriptors, our local voxelizer is based on voxelization of local shape within a cubical volume (rather than a spherical volume) aligned with the LRF, and partitions the volume into uniform bins (i.e., voxels) such that local features inside each bin can be equally weighted and extracted more easily, e.g., surface area which requires mesh clipping. Moreover, we propose various candidate features that can be extracted from each bin, and quantitatively compare them to find the best one for constructing the descriptor. Compared to Ref. [26] that also uses voxelization to construct 3D tensors, our local voxelizer performs voxelization within a unique LRF, and thus enables scan alignment using a single pair of matched points based on aligning the LRFs. As a result, local voxelizer requires less amount of overlap for aligning scans [27].

### 3 Local voxelizer shape descriptor

We take a surface mesh  $S$  as input. If a 3D point

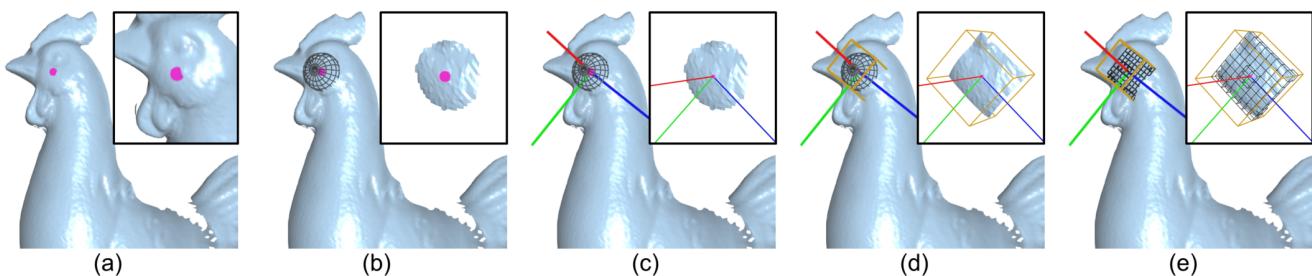
cloud is given, we first convert it into a mesh [28]. A local voxelizer descriptor is a function that assigns to each point  $P \in S$  a vector  $f(P) \in \mathbb{R}^m$  by analyzing the support around  $P$ , where  $m$  is the length of the vector. We first introduce the construction of our local voxelizer and its controlling parameters, then evaluate its performance with respect to mesh noise and varying mesh resolution.

#### 3.1 Local voxelizer construction

Given a basis point  $P$  and a support radius  $r$ , we construct a local voxelizer by defining a unique LRF using the support around  $P$  and by performing local voxelize within the LRF. See Fig. 1. The descriptor vector is calculated by concatenating values computed from shape features (e.g., points, normals, curvatures) of mesh triangles intersecting each voxel.

**Constructing the LRF.** We define the support around  $P$  by intersecting the input mesh  $S$  with a sphere of radius  $r$  centered at  $P$ . See Fig. 1(b). Taking the support as input, we construct an LRF using the method in Ref. [11] which has two steps: (1) construct three orthogonal directions based on principal component analysis (PCA) of triangles in the support; (2) disambiguate the sign of each orthogonal direction to obtain three unique coordinate axes for the LRF. See Fig. 1(c). Note that the sign disambiguation method in Ref. [11] can fail for locally symmetric surfaces, e.g., flat or spherical surfaces. Thus, we use the surface normal of  $P$  to assist in disambiguation: the principal axis associated with the smallest eigenvalue (the red axis in Fig. 1(c)) is in the normal direction.

**Local shape voxelization.** Once the LRF has been constructed, we can define a cubical volume centered at  $P$ , whose edges are aligned with the LRF and have length  $2r$ . See Fig. 1(d). This is the



**Fig. 1** Constructing the local voxelizer. (a) Select a basis point; (b) define support using a sphere centered at the basis point; (c) construct LRF; (d) define a larger local shape using a cube aligned with the LRF; and (e) voxelize the local shape.

smallest cubical volume that contains the support used to construct the LRF. We intersect the cubical volume with the input mesh  $S$ , obtaining a local surface patch  $S_P$ .

Taking  $S_P$  as input, we perform local voxelization for it by partitioning the cubical volume into  $K \times K \times K$  cells (voxels). See Fig. 1(e). For each voxel  $V_i$ , we find the intersection between  $V_i$  and  $S_P$  by clipping  $S_P$  using the six planes of  $V_i$ . The resulting triangles in  $V_i$  are denoted by  $S_P^i$ . For each non-empty voxel, we compute values based on the shape features of  $S_P^i$ , while empty voxels are assigned default zero values. We calculate the descriptor vector by concatenating the values determined for each voxel. Since most of the voxels in the voxelization are likely to be empty, the resulting descriptor will have many zero elements.

**Extracting local feature candidates.** As described above, for each non-empty voxel  $V_i$ , one or more values need to be computed from  $S_P^i$  to represent the local shape in  $V_i$ . In Table 1, we suggest candidate features that can be extracted, as well as the number of output values. For each candidate, we normalize its value(s), e.g.,  $F1, F2, F5$ , and  $F6$  are normalized relative to the voxel size.

Most of the proposed feature candidates are straightforward to calculate except  $F5$  and  $F6$ , and we estimate them using a uniform sampling

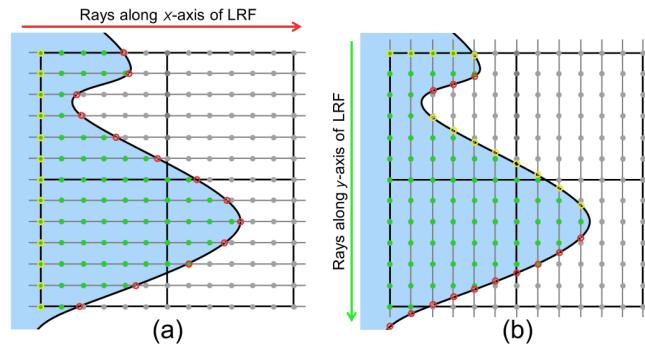
**Table 1** Feature candidates extracted from local shape within each voxel. Note that  $F2$  and  $F6$  use centroid position relative to the minimum point of the voxel cube

No.	Feature	Illustration	Description	# Value
$F1$	Surface area		Total area of clipped mesh triangles in the voxel	1
$F2$	Surface centroid		Average position for local mesh vertices in the voxel	3
$F3$	Average normal		Angle between basis point normal and average normal of local mesh triangles	1
$F4$	Average curvature		Average of surface curvatures for local mesh vertices	1
$F5$	Shape volume		Volume of the local shape enclosed by local mesh and related voxel faces	1
$F6$	Shape centroid		Centroid of the local shape enclosed by local mesh and related voxel faces	3

approach [29]. For a local voxelization with  $K \times K \times K$  voxels, we first build a  $(b \times K + 1)^3$  uniform 3D point grid within it, where we have  $b + 1$  sample points along each edge of each voxel. Then we cast  $(b \times K + 1)^2$  rays through the local mesh  $S_P$ , where each ray passes through  $(b \times K + 1)$  sample points. We compute intersecting points between each ray and  $S_P$ , and identify each intersecting point as inside or outside the object based on the angle between the normal of the point and the ray direction (we set the threshold to  $90^\circ$ ). We classify each sample point as interior or exterior by checking if it is located between the inner intersecting point (or the voxel boundary) and the outer intersecting point along the ray direction. See Fig. 2.  $F5$  is estimated by counting the number of interior sample points and then computing their percentage coverage within the voxel, while  $F6$  is estimated by averaging the positions of all interior sample points. In our experiments, we perform ray casting along the  $x$ -axis of the LRF such that most rays intersect  $S_P$  only once. See Figs. 1(e) and 2(a). The concept behind  $F5$  is similar to the integral volume feature employed in Ref. [7], but using a cubical instead of spherical bounding volume.

### 3.2 Local voxelizer generation parameters

There are three parameters that control local voxelizer descriptors: (i) the support radius  $r$ , (ii) the voxel grid resolution  $K$ , and (iii) the local feature  $f$  extracted for each voxel. We conducted experiments for different settings of parameters using the criterion of recall versus 1-precision curve (RP curve) [30]. See Ref. [27] for details. These experiments indicate that



**Fig. 2** Our local shape volume and centroid estimation method for ray directions along (a)  $x$ -axis and (b)  $y$ -axis of LRF, where  $b = 6$ . Inner and outer ray-mesh intersecting points are marked as yellow and red circles respectively, while interior and exterior sample points are marked in green and gray respectively.

we should select  $r = 0.05d$ ,  $K = 12$ , and  $f = F1$  when generating the descriptors, where  $d$  is the average diagonal size of input scans, since these parameter settings enable the descriptor to have rich descriptiveness and to be generated with reasonable computation cost.

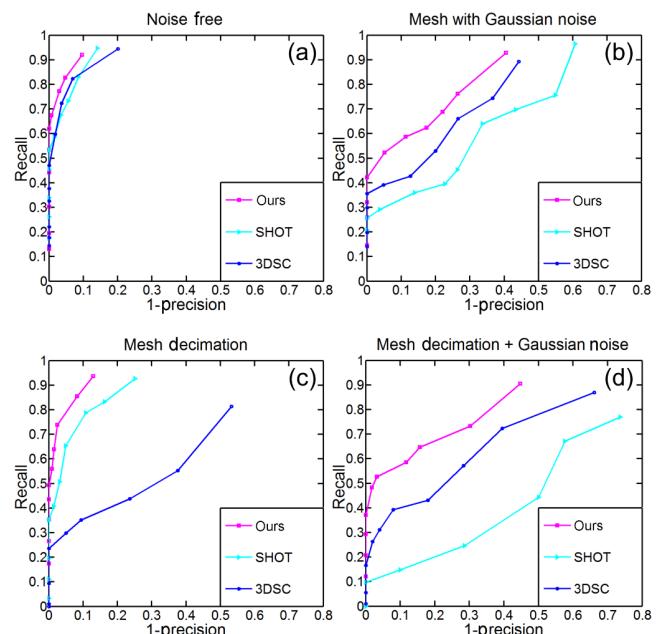
### 3.3 Evaluation of local voxelizer

We now evaluate the robustness of this local voxelizer with respect to mesh noise and varying mesh resolution using the UWA dataset [26] and RP curves. In this experiment, our descriptor was compared to two state-of-the-art descriptors: 3D shape contexts (3DSC) [25] and signature of histograms of orientations (SHOT) [10].

To enable a fair comparison, we adopted the same LRF construction and matching measure for all three descriptors. To construct the descriptor, 3DSC and SHOT both select a spherical support aligned with the LRF centered at a basis point, then partition the support into bins along the radial, azimuth, and elevation directions. For each bin, 3DSC calculates a single value using weighted point density while SHOT calculates a vector of values using a histogram of angles between each point normal and the normal at the basis point. Table 2 shows the parameters that we used for each descriptor after tuning. Note that we did not discard the characteristics of the descriptors that require a specific treatment, e.g., logarithmic partitioning in the radial direction for 3DSC.

We performed the comparison with four variations of the scan data: (1) original data, (2) data with added Gaussian noise ( $0.001d$  standard deviation), (3) data after mesh decimation ( $1/8$  of the original mesh resolution), and (4) data with both noise and decimation.

Figure 3(a) shows that the three descriptors achieved comparable performance on the original data, while our local voxelizer gave the highest recall when 1-precision is close to zero. Figure 3(b) shows that the performance of all descriptors dropped on noisy scan data since the repeatability of LRF and



**Fig. 3** RP curves for the three descriptors in the presence of mesh noise and decimation.

local features extracted is affected by the noise. Among the three descriptors, our local voxelizer performed best since the surface area feature it uses is more robust to noise. Figure 3(c) shows that mesh decimation affects the performance of local voxelizer only slightly since surface area changes little. However, the performance of 3DSC dropped significantly as the point density feature they used becomes less representative as mesh vertices get sparser. Figure 3(d) shows that our local voxelizer again was the best performing method when both mesh noise and decimation were considered.

Our local voxelizer achieved the best performance under all four conditions. However, its efficiency is the lowest of all three since: (1) it requires local mesh clipping to calculate accurate surface areas (i.e., to extract local features) in each voxel, and (2) it has the largest dimensionality, which is an issue when computing, storing, and comparing descriptors. To alleviate the above two issues, we can estimate surface area in each voxel by just using the total area of all triangles intersecting the voxel without local mesh clipping when the mesh is dense. We can also reduce the descriptor dimensionality by using a smaller  $K$  (e.g.,  $K = 8$ ) to speed up the computation, at the expense of sacrificing its descriptive performance somewhat.

**Table 2** Parameter settings for the three descriptors

	Support radius	Bin feature	Dimensionality	Length
3DSC	$0.05d$	Point density	$10 \times 8 \times 16$	1280
SHOT	$0.05d$	Histogram of normals	$8 \times 2 \times 2 \times 10$	320
Ours	$0.05d$	Surface area	$12 \times 12 \times 12$	1728

## 4 Surface registration

So far we have described a novel shape descriptor based on local voxelization within a unique LRF. In this section, we present a pairwise registration algorithm based on this descriptor, and then a surface reconstruction method that can acquire 3D models from a set of scans automatically.

### 4.1 Pairwise registration algorithm

Given a data scan  $S_d$  and a fixed reference scan  $S_r$ , our pairwise registration algorithm consists of four key steps to align  $S_d$  with  $S_r$ : scan representation, generating scan alignment candidates, selecting best scan alignment, and refining scan alignment.

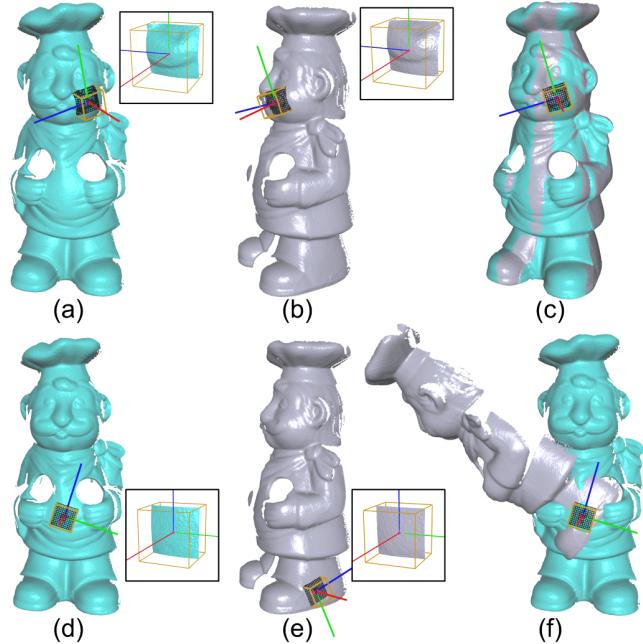
**Step 1: scan representation.** Given a scan, we first select  $N$  seed points from the scan point cloud. To better represent the scan, the seed points should cover the whole scanned surface and should not be too close to each other. Thus, we randomly sample the point cloud and enforce minimal separation distance between the samples to obtain  $N$  seed points. For each seed point, the corresponding LRF and local voxelizer are constructed, and stored in a library. We selected  $N = 2000$  in experiments as a tradeoff between computational cost and sampling performance.

To align a pair of scans  $S_d$  and  $S_r$ , we could simply match the descriptors of their sampled seed points. However, since the seed points cover the whole scan surface evenly, one randomly picked point on  $S_d$  could match one seed point on  $S_r$  correctly, given that the physical distance between the seed point and the real match on  $S_r$  is small. In our experiment, we find that sampling  $M = 200$  feature points on  $S_d$  and matching their descriptors with those of  $N$  seed points on  $S_r$  can provide good matching results, and vice versa. Thus, for each scan, we further sample  $M$  feature descriptors from the original  $N$  seed descriptors, and store them in the library.

**Step 2: generating scan alignment candidates.** To align  $S_d$  with  $S_r$ , each feature descriptor of  $S_d$  is compared to all seed descriptors of  $S_r$ . If the Euclidean distance between the two descriptor vectors is less than a threshold, the feature point on  $S_d$  and its closest seed point on  $S_r$  are considered a match. However, this match is not guaranteed to be correct since: (1) there could be no or very small overlap between  $S_d$  and  $S_r$ ; (2) the

local shape around the feature point may not be discriminative, e.g., having flat or spherical shape; and (3) there may exist similar or symmetrical shape features on  $S_r$ . Each generated match leads to a scan alignment candidate (i.e., a  $4 \times 4$  transformation matrix), found by aligning the three axes of the uniquely defined LRFs. See Fig. 4 for two examples.

**Step 3: selecting best scan alignment.** By matching the descriptors of  $S_d$  and  $S_r$ , we can obtain about  $M$  alignment candidates. We first sort these candidates based on descriptor distance and then pick the best five candidates with smallest distance. To find the true best of these five candidates, we evaluate each candidate by transforming  $S_d$  into  $S_r$  and estimating the normalized overlap area between the transformed  $S_d$  (denoted by  $S_d^t$ ) and  $S_r$ . In detail, we first build a  $kd$ -tree for the point cloud of  $S_r$ , then for each point in  $S_d^t$ , find its closest point in  $S_r$ . We consider a point in  $S_d^t$  and its closest point in  $S_r$  as overlapping if their distance is smaller than a threshold. The score of the alignment candidate is calculated as the overlap area divided by the surface area of  $S_d$ . Lastly, we select as the output the candidate with the highest score from the five candidates.

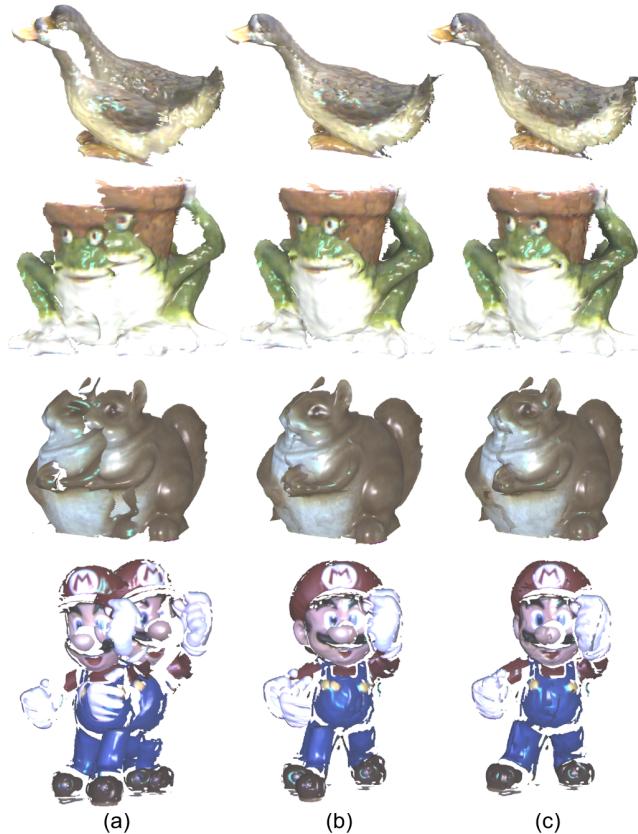


**Fig. 4** Aligning two scans of CHEF by matching local voxelizers. (a) and (b) A pair of correctly matched descriptors (zoom-in views show local shape); (c) scans aligned based on the LRFs; (d) and (e) another pair of matched descriptors; (f) scans not properly aligned as the selected local shape is flat and not discriminative.

**Step 4: refining scan alignment.** Using the above procedure, a pair of scans with a certain amount of overlap can always be properly aligned. We further refine the alignment by using ICP [3]. Since the initial transformation calculated by aligning LRFs is very close to the correct alignment, near perfect alignment can be achieved using only one or two ICP iterations. Figure 5 shows our pairwise alignment results on four objects from the Bologna dataset [10].

#### 4.2 Model surface reconstruction

Given a set of  $n$  input scans with arbitrary initial poses, we could register them automatically by directly applying the above pairwise algorithm. However, this will result in a high computation cost  $O(n^2m)$ , where  $m$  is the average time taken for each pairwise alignment. The reason is that two randomly selected scans could share no or little overlap, and the pairwise algorithm cannot generate a correct alignment for such case. Thus, we may need to loop



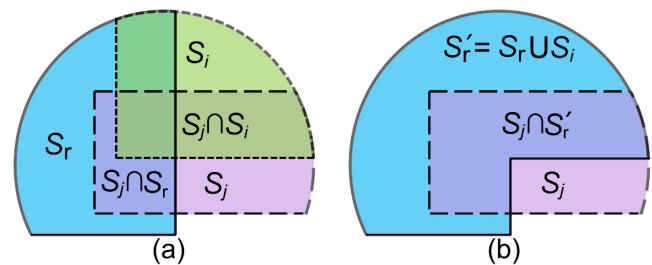
**Fig. 5** Pairwise alignment results for four objects from the Bologna dataset: DUCK, FROG, SQUIRREL, and SUPER MARIO: (a) two input scans; (b) scans aligned by matching local voxelizer and aligning corresponding LRFs; and (c) refined alignment by applying ICP.

through all possible scan pairs to register the input scans successfully.

To reconstruct model surface from input scans more efficiently, we propose an expanding strategy based on merging and expanding the descriptor representation of aligned scans. We first select a scan with the largest surface area as a reference scan  $S_r$ , then we select one of the remaining  $n - 1$  scans, denoted as  $S_i$  and align it with  $S_r$  using the pairwise algorithm, until the alignment score is larger than a threshold (e.g., 0.5). By refining the alignment of  $S_i$  and  $S_r$  using ICP, we can merge the two scans and their descriptor representations accurately, thus obtaining a larger scan  $S'_r$ . To reduce the number of merged descriptors for matching, we remove duplicated descriptors within the overlap region of  $S_r$  and  $S_i$  by measuring the physical distance between their seed points.

We further select an additional scan  $S_j$  from amongst the remaining  $n - 2$  scans and align  $S_j$  with  $S'_r$  using the same pairwise algorithm, until the alignment score is larger than the threshold. Since  $S'_r$  covers a larger portion of object surface, the chance of successfully aligning  $(S_j, S'_r)$  should be larger than that of aligning  $(S_j, S_r)$  or  $(S_j, S_i)$ . See Fig. 6. Moreover, the descriptor matching time is also reduced since we avoid matching duplicated descriptors in the overlap of  $S_r$  and  $S_j$ . We iterate the above procedure until all input scans have been registered and included in  $S'_r$ . In our experiments, we found that after registering a few scans, so that  $S'_r$  covers a large portion of the 3D object surface, one or two trials are enough to align each of the remaining scans with  $S'_r$ .

Once all input scans have been brought into alignment, we create a large oriented point cloud by summarizing the point cloud (with normals) of each



**Fig. 6** Overlapping regions of (a)  $(S_j, S_r)$ ,  $(S_j, S_i)$  and (b)  $(S_j, S'_r)$ . Here,  $S_r$ ,  $S_i$ , and  $S_j$  are differentiated by boundary style and filling color (blue, green, and purple).

registered scan, and reconstruct a mesh model using Poisson surface reconstruction [31]. Our expanding strategy along with the pairwise algorithm based on our local voxelizer makes our approach especially suitable for registering scans with small overlap. As a result, we can reconstruct a 3D model from a small number of input scans, saving effort during both scan capture and registration. Our approach can acquire a complete 3D model from only 8 scans (see for example CHICKEN and CHEF models in Fig. 7).

## 5 Results

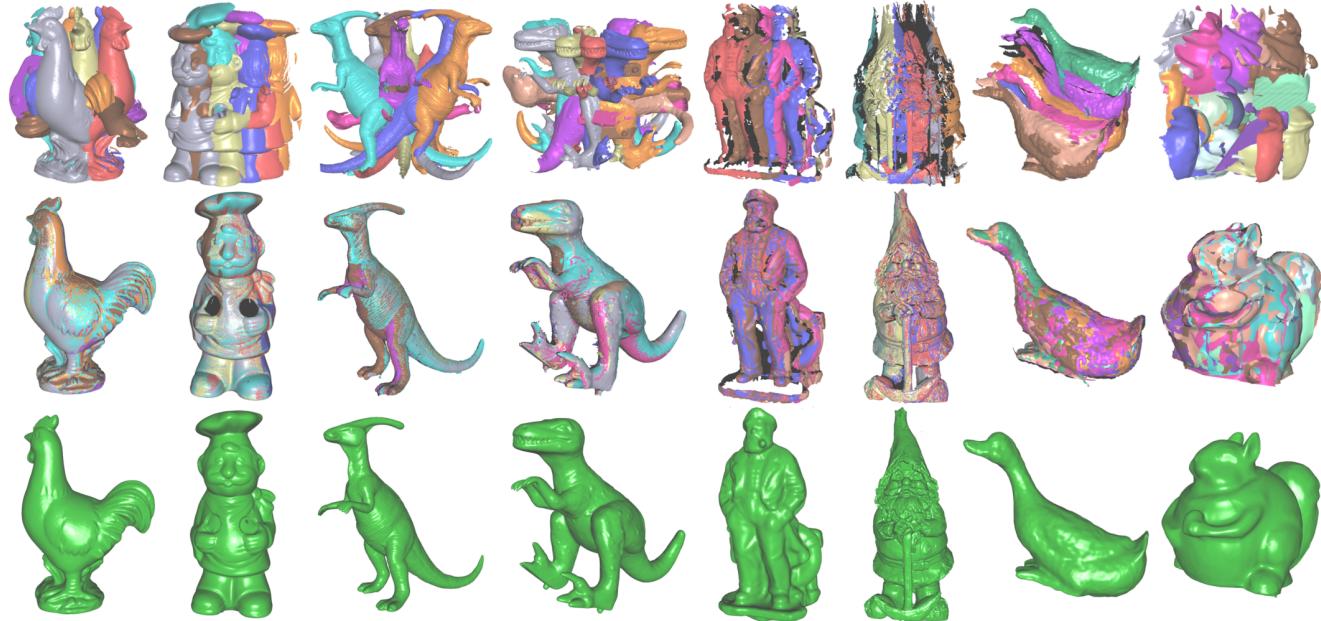
We have tested our surface registration approach using three publicly available datasets: (1) four objects from the UWA dataset [26] (CHEF, CHICKEN, PARASAUROLOPHUS, and T-REX), (2) two objects from the Queen’s dataset [32] (SAILOR and GNOME), and (3) two objects from the Bologna dataset [10] (DUCK and SQUIRREL).

Since our method supports reconstructing a complete 3D model from a small number of scans, we manually sampled about 10 scans for each object to speed up the registration process; these selected scans cover most of the object’s surface. These selected scans were used as input to reconstruct a 3D model.

**Statistics and performance.** We implemented

our method in C++ and executed it on a desktop PC with an Intel i7-3770 CPU (3.4 GHz, 4 cores) and 8 GB memory. Table 3 presents statistics concerning our results, including number of input scans, average number of triangles in each scan, time to calculate descriptor representations, time to register all scans, time to reconstruct 3D model surface, and total time taken (see Fig. 7 for the scans and models). In general, our method takes a few minutes to produce a 3D model from around 10 scans, where the number of triangles in each scan ranges from 10,000 to 300,000. The scan description procedure takes most of the computation time since we need to calculate around 2000 descriptors for each scan, and each descriptor calculation includes LRF construction, local mesh clipping, voxelization, etc.

**Reconstructed 3D models.** Figure 7(top) shows input scans of various 3D objects in default poses (translated for ease of visualizing their shapes). Figure 7(middle) shows the scans registered by our method. While some scans cover a small portion of the object’s surface and have small overlap with others (e.g., see SAILOR), our method still can register them successfully. Figure 7(bottom) shows 3D models of varying shape complexity reconstructed from the aligned scans using the method in Ref. [31]. Since Ref. [31] fills holes on the model surface automatically when there is lack



**Fig. 7** 3D models reconstructed using our method. Top to bottom: input scans, registered scans, and reconstructed 3D model. Left to right: CHICKEN, CHEF, PARASAUROLOPHUS, T-REX, SAILOR, GNOME, DUCK, and SQUIRREL.

**Table 3** Statistics for the registration results in Fig. 7

	Number of scans	Average number of triangles	Description time (min)	Registration time (min)	Reconstruction time (min)	Total time (min)
CHICKEN	8	51,599	2.20	0.11	0.31	2.62
CHEF	8	135,596	3.42	0.13	0.62	4.17
PARASAUROLOPHUS	9	67,887	2.53	0.22	0.37	3.12
T-REX	10	64,237	3.07	0.21	0.38	3.66
SAILOR	10	11,073	0.91	0.16	0.09	1.16
GNAME	9	285,432	7.37	0.73	1.84	9.94
DUCK	12	46,785	2.43	0.38	0.27	3.08
SQUIRREL	16	91,546	5.94	0.43	0.98	7.35

of point cloud data (see the two big holes in CHEF), the reconstructed 3D model differs from the original object in these regions. However, this issue can be resolved by employing other surface reconstruction techniques.

## 6 Conclusions

This paper has presented a novel *local voxelizer* descriptor for surface registration. This local voxelizer is constructed by defining a unique LRF for the support around a basis point and by performing local voxelization within the LRF. A surface registration approach is given based on the descriptor and an expanding strategy that merges descriptor representations of aligned scans, enabling it to register scans with small overlap. Experiments show that local voxelizer is robust to mesh noise and varying mesh resolution, and our registration approach allows acquisition of complete 3D models of various 3D objects from only a few scans.

**Future work.** Firstly, we currently convert an input point cloud into a mesh before registration, in which useful information could be lost. Thus, we plan to extend our descriptor and registration approach to directly work on (noisy) point cloud data. Secondly, we plan to extend our descriptor to combine shape features with photometric features such as silhouettes [33] and textures [34] in 2D images to enrich local surface description.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 61403357), Anhui Provincial Natural Science Foundation (No. 1508085QF122), and Fundamental Research Funds for the Central Universities (No. WK0110000044).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

- [1] Newcombe, R. A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A. J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In: Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality, 127–136, 2011.
- [2] Mellado, N.; Aiger, D.; Mitra, N. J. Super 4PCS fast global pointcloud registration via smart indexing. *Computer Graphics Forum* Vol. 33, No. 5, 205–215, 2014.
- [3] Besl, P. J.; McKay, N. D. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 14, No. 2, 239–256, 1992.
- [4] Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image and Vision Computing* Vol. 10, No. 3, 145–155, 1992.
- [5] Chen, C.-S.; Hung, Y.-P.; Cheng, J.-B. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 21, No. 11, 1229–1234, 1999.
- [6] Mitra, N. J.; Gelfand, N.; Pottmann, H.; Guibas, L. Registration of point cloud data from a geometric optimization perspective. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 22–31, 2004.
- [7] Gelfand, N.; Mitra, N. J.; Guibas, L. J.; Pottmann, H. Robust global registration. In: Proceedings of the 3rd Eurographics Symposium on Geometry Processing, Article No. 197, 2005.
- [8] Aiger, D.; Mitra, N. J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics* Vol. 27, No. 3, Article No. 85, 2008.

- [9] Liu, Y.; Zhou, W.; Yang, Z.; Deng, J.; Liu, L. Globally consistent rigid registration. *Graphical Models* Vol. 76, No. 5, 542–553, 2014.
- [10] Tombari, F.; Salti, S.; Stefano, L. D. Unique signatures of histograms for local surface description. In: Proceedings of the 11th European Conference on Computer Vision: Part III, 356–369, 2010.
- [11] Guo, Y.; Sohel, F.; Bennamoun, M.; Lu, M.; Wan, J. Rotational projection statistics for 3D local surface description and object recognition. *International Journal of Computer Vision* Vol. 105, No. 1, 63–86, 2013.
- [12] Zhong, Y. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In: Proceedings of IEEE 12th International Conference on Computer Vision Workshops, 689–696, 2009.
- [13] Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In: Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling, 145–152, 2001.
- [14] Brown, B. J.; Rusinkiewicz, S. Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics* Vol. 26, No. 3, Article No. 21, 2007.
- [15] Bouaziz, S.; Tagliasacchi, A.; Pauly, M. Sparse iterative closest point. *Computer Graphics Forum* Vol. 32, No. 5, 113–123, 2013.
- [16] Pottmann, H.; Huang, Q.-X.; Yang, Y.-L.; Hu, S.-M. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal of Computer Vision* Vol. 67, No. 3, 277–296, 2006.
- [17] Cheng, Z.-Q.; Chen, Y.; Martin, R. R.; Lai, Y.-K.; Wang, A. SuperMatching: Feature matching using supersymmetric geometric constraints. *IEEE Transactions on Visualization and Computer Graphics* Vol. 19, No. 11, 1885–1894, 2013.
- [18] Gal, R.; Cohen-Or, D. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics* Vol. 25, No. 1, 130–150, 2006.
- [19] Chua, C. S.; Jarvis, R. 3D free-form surface registration and object recognition. *International Journal of Computer Vision* Vol. 17, No. 1, 77–99, 1996.
- [20] Huang, Q.-X.; Flöry, S.; Gelfand, N.; Hofer, M.; Pottmann, H. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics* Vol. 25, No. 3, 569–578, 2006.
- [21] Pottmann, H.; Wallner, J.; Huang, Q.-X.; Yang, Y.-L. Integral invariants for robust geometry processing. *Computer Aided Geometric Design* Vol. 26, No. 1, 37–60, 2009.
- [22] Albarelli, A.; Rodolà, E.; Torsello, A. Loosely distinctive features for robust surface alignment. In: *Lecture Notes in Computer Science*, Vol. 6315. Daniilidis, K.; Maragos, P.; Paragios, N. Eds. Berlin Heidelberg: Springer, 519–532, 2010.
- [23] Johnson, A. E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 21, No. 5, 433–449, 1999.
- [24] Huber, D. F.; Hebert, M. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing* Vol. 21, No. 7, 637–650, 2003.
- [25] Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; Malik, J. Recognizing objects in range data using regional point descriptors. In: *Lecture Notes in Computer Science*, Vol. 3023. Pajdla, T.; Matas, J. Eds. Berlin Heidelberg: Springer, 224–237, 2004.
- [26] Mian, A. S.; Bennamoun, M.; Owens, R. A. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal of Computer Vision* Vol. 66, No. 1, 19–40, 2006.
- [27] Song, P.; Chen, X. Pairwise surface registration using local voxelizer. In: *Proceedings of Pacific Graphics*, Stam, J.; Mitra, N. J.; Xu, K. Eds. The Eurographics Association, 1–6, 2015.
- [28] Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. In: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, 71–78, 1992.
- [29] Song, P.; Fu, Z.; Liu, L.; Fu, C.-W. Printing 3D objects with interlocking parts. *Computer Aided Geometric Design* Vols. 35–36, 137–148, 2015.
- [30] Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 27, No. 10, 1615–1630, 2005.
- [31] Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson surface reconstruction. In: Proceedings of the 4th Eurographics Symposium on Geometry Processing, 61–70, 2006.
- [32] Taati, B.; Greenspan, M. Local shape descriptor selection for object recognition in range data. *Computer Vision and Image Understanding* Vol. 115, No. 5, 681–694, 2011.
- [33] Song, P.; Wu, X.; Wang, M. Y. A robust and accurate method for visual hull computation. In: Proceedings of International Conference on Information and Automation, 784–789, 2009.

- [34] Song, P.; Wu, X.; Wang, M. Y.; Wu, J. Expansion-based depth map estimation for multi-view stereo. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 3213–3218, 2010.



**Peng Song** is an associate professor in School of Computer Science and Technology, University of Science and Technology of China. He received his B.S. degree from Harbin Institute of Technology (2007), M.S. degree from Harbin Institute of Technology Shenzhen Graduate School (2010), and Ph.D. degree from Nanyang Technological University,

Singapore (2013). His research interests lie on computer graphics, computer vision, and human-computer interaction.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.