Special Issue on SMI 2016

Color-aware surface registration

Shuai Lin [a], Yu-Kun Lai [b], Ralph R. Martin [b], Shiyao Jin [a], Zhi-Quan Cheng [c],*

[a] School of Computer, National University of Defense Technology, Changsha City, Hunan Province 410073, China
[b] School of Computer Science and Informatics, Cardiff University, Cardiff, Wales CF24 3AA, UK
[c] Avatar Science Company, Changsha City, Hunan Province 410205, China

ABSTRACT

Shape registration is fundamental to 3D object acquisition; it is used to fuse scans from multiple views. Existing algorithms mainly utilize geometric information to determine alignment, but this typically results in noticeable misalignment of textures (i.e. surface colors) when using RGB-depth cameras. We address this problem using a novel approach to color-aware registration, which takes both color and geometry into consideration simultaneously. Color information is exploited *throughout* the pipeline to provide more effective sampling, correspondence and alignment, in particular for surfaces with detailed textures. Our method can furthermore tackle both rigid and non-rigid registration problems (arising, for example, due to small changes in the object during scanning, or camera distortions). We demonstrate that our approach produces significantly better results than previous methods.

© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Reconstructing 3D objects from multiple scans taken from different viewpoints is a classical problem. A fundamental step in this task is surface registration, in which geometric data in local coordinate systems are aligned to a global coordinate system. Registration involves two intertwined problems: establishing correct correspondences between points on different surfaces, and finding a suitable spatial transformation or deformation that puts these surfaces into alignment. This is a computationally expensive task; most methods iteratively optimize correspondences and transformations alternately.

Surface registration has made substantial progress, and existing methods perform quite well, even in challenging cases with little overlap, noise and outliers. However, many methods formulate registration as an optimization problem based on *geometric* errors, and only use geometric information for finding correspondences and transformations. Such methods work well with textureless surfaces (in this paper, *texture* refers to surface coloring, not shape detail). However, recently developed low-cost RGB-depth (RGB-D) acquisition devices such as the Microsoft Kinect permit efficient and cheap capture of textured surfaces, leading to many novel applications such as clothed 3D human body reconstruction. Often, the color information has greater detail than the geometric information, so registration based on geometry alone can lead to

poor results in which the textures are not well aligned. Worse still, texture misalignments are typically much more noticeable to the eye than geometric misalignments.

To overcome this problem, we present a novel color-aware registration algorithm that produces high-quality registration of textured surfaces. Our method can handle both rigid and non-rigid alignment: even if scans are supposed to be rigidly related, non-rigid alignment may be needed to correct for inaccurately determined camera intrinsic parameters, to allow for lens distortion, or to rectify small geometric changes (e.g. altered wrinkles in clothing when a human subject has moved slightly between successive scans).

Our first contribution is to use color information as well as geometric information to robustly find correspondences, in both rigid and non-rigid cases. We filter out incorrect vertex correspondences by using a combination of color, texture and geometric measures, and further improve pruning by rejecting correspondences which are not mutually consistent. Our second contribution is to also take color into account when using optimization to find the transformation or deformation needed for surface alignment.

We experimentally evaluate our *color-aware registration algorithm* using real scans, demonstrating the effectiveness of our algorithm when the subjects have richly textured clothing. We also compare our algorithm with other state-of-the-art methods using both real scans and a public dataset, showing the superiority of our method. While we have mainly tested the algorithm on our own data, we believe it to be generally useful, as identical problems are likely to arise in other RGB-D capture systems when capturing subjects with rich color textures.

* Corresponding author.
E-mail address: cheng.zhiquan@avatarscience.com (Z.-Q. Cheng).

2. Related work

Surface registration is an active research area. Various surveys [1–3] have summarized and categorized methods from different perspectives, including performance of descriptors, rigid versus non-rigid registration framework, etc. Here we only summarize the work most relevant to our problem.

2.1. Registration based on geometric closeness

Registration aims to put two (or more) shapes into alignment, so it is natural to use geometric closeness as a criterion to determine the transformations. In the rigid case, the most popular approach is the *iterative closest point* (ICP) method [4,5]. The main ideas are point correspondence selection based on Euclidean distances, and an energy function formulated based on closeness, either *point-to-point* or *point-to-plane*, for transformation optimization. Its many variants address different challenges, including noise, outliers, limited amounts of overlap, variations in initial positions, etc. Recent improvements have been achieved by care in correspondence selection [6], using L^p distances in the energy formulation [7], and use of branch-and-bound [8] or game theory [9] to explore the motion space. The approach has also been generalized largely unchanged to non-rigid registration, using dense reliable correspondences to find optimal local transformations [10–13]. To address the high degrees of freedom, priors are required, such as smoothness and piecewise rigidity.

2.2. Features, salience and texture assisted registration

While ICP approaches are effective, good initialization is required to avoid such iterative methods stopping in a local minimum. As an alternative, feature-based approaches identify correspondences based on intrinsic geometric features. This helps us to identify potential correspondences even when the shapes are far from alignment and reduces the possibility of mismatches. Features used include spin images [14], mean [15] and Gaussian [16] curvature, SHOT signatures [17], and salience measures such as differential properties [15] or multiscale slippage [18]. Color at each vertex can also be used to provide features, either as direct color values [19–21] or by derivation from color values, e.g. texture spin images [22] and orbital point descriptors [23]. Both [24] and [25] use color to assist registration and are thus closely related to our paper. The former uses SURF descriptors to help find correspondences and incorporates color information into the energy representing alignment error. However, their focus is completion and reconstruction of dynamic shapes from real-time data. Thus, their SURF descriptor is augmented with a temporal coherence constraint, which is not relevant in our setting where the multiple views differ substantially both in time and space. The latter focuses on texture correction to improve shape and texture reconstruction. However, their method aims to reconstruct static, rigid objects. Thus, their method starts by reconstructing a textureless model using KinectFusion to establish uniform geometric constraints, and then optimizes texture consistency locally. Although they consider sensor distortion, and apply non-rigid correction to the images, they do not change the geometry provided by KinectFusion. The method thus cannot cope with the data we assume, where as noted, multiple scans can differ substantially from views and minor non-rigid deformations are assumed to be present: KinectFusion reconstruction does not allow for such situations. The strength of our approach is to *combine* geometry and texture information during the whole process, allowing us to find good alignment even in challenging cases.

2.3. Other uses of color and geometric descriptors

Color has also been exploited in addition to geometric information in other applications such as shape retrieval and recognition. Various effective descriptors have been proposed, including heat kernel signatures [26], conformal factors [27], SIFT features [28], wave kernel signatures [29], and MeshHOG (histograms of oriented gradients) [30]. Although intended for applications which only need sparse correspondences, such descriptors have the potential for use in shape registration. In this paper, we use MeshHOG [30] during correspondence search.

Several works [31–33] have considered optimization of textures on a 3D mesh. Multi-view textures can be projected onto a pre-built 3D mesh and rectified to achieve texture consistency via feature correspondences, using e.g. SIFT [31] and optical flow [32], or discrete labeling optimization on each triangle [33]. Although such methods achieve consistent colorization for a 3D mesh, which is one of our goals as well, they assume rather simple geometry which can be reconstructed using existing registration methods, avoiding the difficulty of having to simultaneously optimize both geometry and texture. As Fig. 11 shows later, for challenging situations, independently performing shape registration to initially build a 3D mesh does not work well, whereas our color-aware surface registration exploits both color and geometry uniformly at all stages, producing substantially better results.

2.4. Textured surface fusion

Rapid advances in RGB-D cameras have driven applications based on 3D modeling of highly textured surfaces. The GPU-based capture system proposed in [34] uses a single Kinect camera to incrementally build 3D models. Tong et al. [35] use three Kinect sensors and associated components to produce acceptable 3D human body models. A fixed Kinect can be used with user orientation changes to capture and produce a full-body 3D self-portrait [36]. These systems all use traditional registration techniques to fuse scans. However, data captured by cheap RGB-D cameras has significant geometric distortions due both to camera distortion and errors introduced by depth processing; multiple frames captured even from a perfectly static subject may not align perfectly. If a human subject is captured, misalignment is further exacerbated by small unintentional movements. Our color-aware registration algorithm utilizes color information both when finding correspondences and when improving alignment; the emphasis is on providing a high-quality texture on the recovered surface.

The method in [37] formulates registration in an overall optimization framework which requires solution of a nonlinear system. This is difficult, so local linearization is resorted to. The method is furthermore used in a face tracking problem where strong priors are available. The blendshape model used is a linear combination of coordinates and is inapplicable to general objects such as human bodies. Other recent work also considers real-time non-rigid reconstruction from RGB-D data. The method in [38] is effective but requires a complete static template model to be captured first; this template is deformed to fit a real-time stream of scans. Alternatively, [39] does not require a template model, and demonstrates the effectiveness of using *sparse* color feature points in a tracking scenario (where adjacent frames are very similar) to help improve registration. Both methods require adjacent frames to be highly similar, so are unsuitable for non-rigid registration of general pairs of scans as considered in our problem. Our method also differs in that *dense* color information is used to align two scans to improve the geometry and *texture* alignment. Although having a different purpose, RGB-D SLAM (simultaneous localization and mapping) also considers use of RGB-D data to reconstruct 3D scenes [40,41]. Both color and geometry information are used

to simultaneously estimate camera pose and fuse point clouds. However, again, they focus on *online* applications where adjacent frames are very similar.

Here, we extend existing registration frameworks (ICP [4] for rigid registration, and embedded graphs [10] for the non-rigid case), and use color descriptors [30,42] to improve correspondence determination as well as to guide alignment. Unlike such previous registration methods, we use both geometric and texture information for optimization, and use multiple filters to search for correct correspondences between data captured from different viewpoints. After finding correspondences based on geometry and color—good matches must respect both—our method recovers an alignment result which provides a good compromise between texture consistency and geometric consistency, in both rigid and non-rigid cases. Our method utilizes color information in both rigid and non-rigid registration in a consistent manner, and produces robust results by using effective descriptors.

3. Color aware registration

We now explain our algorithm for accurate registration which exploits color information. We consider pairwise registration for both rigid and non-rigid cases.

The following notation is used. A textured mesh surface S has a set of vertices $V = \{v\}$ and a set of faces $F = \{f\}$; N is the number of vertices. Pairwise registration works on two partially overlapping surfaces $S^a = \{V^a, F^a\}$ and $S^p = \{V^p, F^p\}$, where a stands for active and p stands for passive. During registration, transformations or deformations are used to align the active surface with the passive one. $\langle v_i^a, v_i^p \rangle$ represents a pair of corresponding vertices, one from each surface. We assume that S^a and S^p have initially been coarsely aligned, e.g. from knowledge of the camera configuration. A rigid transformation $T = (R, t)$ comprises a rigid rotation R and translation t.

3.1. Pairwise color-aware rigid registration

Rigid registration involves finding a global transformation with 6 degrees of freedom (DoF). For surfaces which also undergo minor deformation, it is appropriate to factor out a rigid transformation before performing non-rigid registration; the rigid alignment provides good initialization for non-rigid alignment. We thus first consider rigid registration. Given the few DoF, three corresponding vertex pairs suffice to uniquely determine a rigid transformation T. Further correspondences may help us to improve robustness, but may also introduce mismatches that negatively impact the result. The quality of correspondences is more important than their quantity. We conservatively select only those correspondences in which we have high confidence. As we assume that a coarse alignment is given and partial overlaps exist between surfaces, we search in the overlapping areas for correspondences. For both quality of results and efficiency, we sample both surfaces and only match sampled vertices. Vertices with salient colors (i.e. locally distinctive colors) should have a higher probability of being sampled as they are more likely to provide accurate correspondences. Fig. 2 shows our rigid registration pipeline. We give details below.

3.1.1. Sampling candidate vertex sets

We now consider the first three steps in our pipeline. We establish correspondences using subsets of the overlapping parts of V^a and V^p. These subsets are called the *candidate vertex sets* Ω^a and Ω^p. For each vertex in S^a, we find the K closest vertices v_k^p in S^p, and only keep those vertices in S^a that are sufficiently close to S^p in Euclidean space. Instead of setting a threshold, we use a ranking approach which is more robust and insensitive to variations in hardware configuration and other changes. We use the mean value of Euclidean distance between vertex v^a and its K closest vertices v_k^p as the metric in the ranking process: only the closest fraction η of vertices in S^a are preserved as the overlapping area on S^a. Then the overlapping area is used as a sampling pool for S^a. The use of K nearest neighbors and η help improve robustness and efficiency; our experiments suggest that $K = 20$ and $\eta = 75\%$ give a good balance between coverage and efficiency. kD-trees are built to accelerate nearest neighbor searches throughout our algorithm. The same approach is applied to obtain the overlapping area on V^p.

For each vertex in each sampling pool, we first estimate its color distinctness from its neighbors as M_{color}^i and assign it a sampling weight W_{smp}^i:

$$M_{color}^i = \sum_{j \in N_i} \text{SingleColorDist}(v_i, v_j) / |N_i|,$$

$$W_{smp}^i = M_{color}^i / \sum M_{color}^i, \tag{1}$$

where N_i are the indices of the 1-ring neighbors of v_i, and SingleColorDist$(\cdot, \cdot)$ is the L^2 norm between the colors of two vertices, so M_{color}^i represents the mean color difference between v_i and its 1-ring neighbors.

The overlapping areas may contain thousands of vertices. To improve efficiency without visible degradation of results, we randomly sample 5% of these vertices to get Ω^a and Ω^p. The probability that a vertex i is selected is given by W_{smp}^i. Vertices with more salient colors are more likely to be selected, while at the same time, some less salient vertices are still chosen to ensure robustness against outlying vertices in color space. Fig. 3 demonstrates this step.

3.1.2. MeshHOG descriptor

The HOG descriptor for 2D images captures local characteristics of pixels, and has been used to match pixels between transformed images. MeshHOG extends it from 2D to 3D [30], allowing extraction of local characteristics of scalar functions defined at each vertex on a 3D surface. We use MeshHOG to extract local
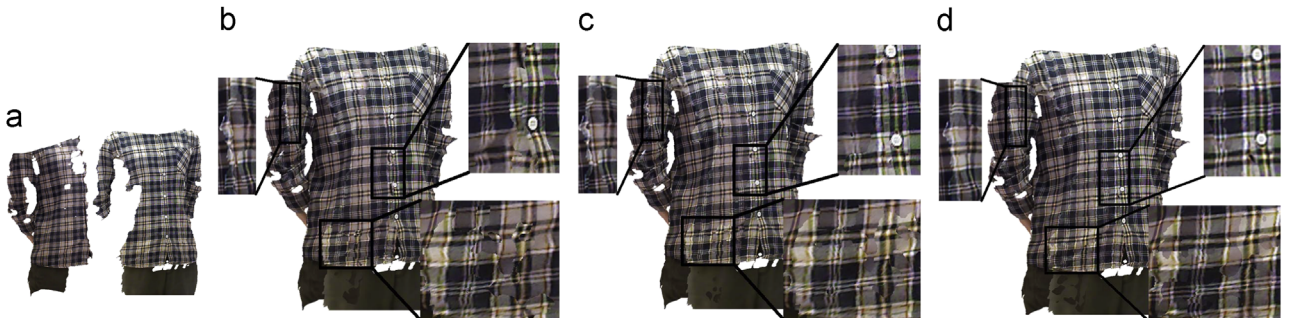
Fig. 1. Using rigid and non-rigid registration to correct misalignments in geometry and texture. Two input textured surfaces (a) are captured by RGB-D cameras; the camera configuration provides initial alignment (b). Successive rigid (c) and non-rigid (d) steps improve it, giving a final surface with high-quality textures.
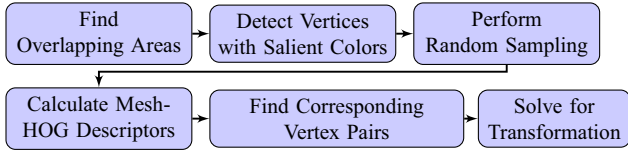
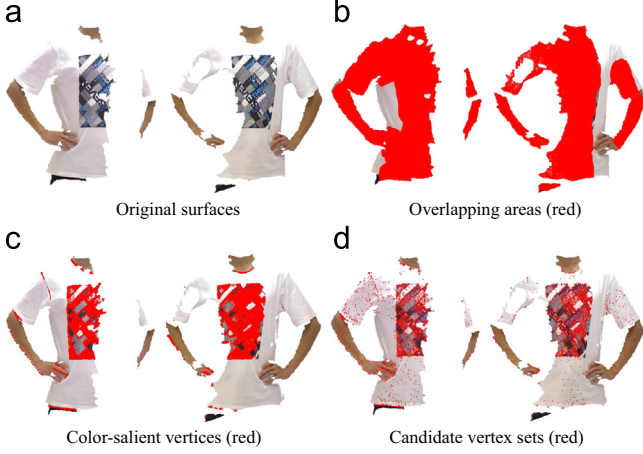**Fig. 2.** Color-aware rigid registration pipeline.

Fig. 3. Candidate vertex selection.

descriptors for the vertices in the *candidate vertex sets*, which are then used for finding correspondences.

3.1.3. Establishment of correspondences

To find good correspondences between S^a and S^p, we first identify candidate vertex pairs by finding the K_{rigid} closest vertices from Ω^p to each vertex in Ω^a. We use cascaded filtering for efficiency. Candidate pairs are ranked according to their Euclidean distance and their color distance (in the L^2 norm). Only vertex pairs which are in the best fraction κ of both ranking arrays are retained (choice of κ will be discussed later). These are then further pruned to obtain one-to-one vertex pairs as follows. Suppose that for v_i^a from Ω^a, vertex pair $\langle v_i^a, v_j^p \rangle$ has the minimum Mesh-HOG feature distance of all remaining vertex pairs $\langle v_i^a, \cdot \rangle$ related to v_i^a, $\langle v_i^a, v_j^p \rangle$ is retained temporarily. Then we swap roles of Ω^a and Ω^p and search again as above, and check whether v_i^a is the vertex which has minimum MeshHOG feature distance from v_j^p. If there is reciprocal agreement, $\langle v_i^a, v_j^p \rangle$ is viewed as a valid correspondence.

3.1.4. Rigid registration

After finding correspondences, we determine the transformation T between the surfaces by minimizing the sum of weighted Euclidean distances between all remaining vertex pairs $\langle v_i^a, v_j^p \rangle$:

$$\arg \min_T \sum_{\langle v_i^a, v_j^p \rangle \in P} w_{ij} \| T \cdot v_i^a - v_j^p \|^2, \tag{2}$$

where the weight $w_{ij} = 1 - \text{SingleColorDist}(v_i^a, v_j^p)/\delta_C$, and δ_C is the maximum color distance observed in the previous color distance ranking. This approach gives greater weight to vertex pairs with more consistent colors. Note that in this energy formulation, once the correspondences have been established, the color-related weights w_{ij} do not change with transformation T. Thus, this energy can be very efficiently optimized, as in standard ICP registration.

The rigid registration process terminates when the transformation matrix relating consecutive registration steps is close to a unit matrix or the number of iterations exceeds a maximum predefined value (which we take to be 50). In detail, if $\|R - I\|_F$ is below 10^{-3} and the magnitude of translation is smaller than the resolution of the current mesh surface, then iteration terminates; here $\| \cdot \|_F$ represents the Frobenius norm of a matrix, R is the rotation matrix, and I is a unit matrix. Fig. 11 shows example results of our rigid registration algorithm.

3.2. Pairwise color-aware non-rigid registration

Rigid registration helps put both surfaces into alignment. However, as the top row of Fig. 12 shows, some misalignments may still remain, especially in boundary areas of each scan. Causes are various: systematic distortion produced by the depth cameras, mutual interference when using multiple infrared projectors, depth processing artifacts caused for example by smoothing, a human subject cannot hold perfectly still during scanning. All lead to non-rigid deformation between scans. To overcome this problem and achieve better alignment results, we further use a non-rigid registration step, again taking color information into account.

Non-rigid registration has many more DoF than rigid registration. Local transformations must be determined; a dense set of correspondences is required. This is much more challenging than rigid registration case for which only a few reliable correspondences suffice. We adapt various techniques to improve correspondences in this case.

The steps used for non-rigid registration are shown in Fig. 4. We first build an embedded graph on each surface, following [10], and use it to control deformation. Local transformations are defined at each node of the embedded graph, with the transformation at each vertex being obtained by a weighted average of transformations at neighboring nodes. Each weight is proportional to the inverse of the distance between the vertex and the node.

3.2.1. Random candidate sampling

As in rigid registration, we start by randomly sampling vertices to provide a candidate set. For efficiency, correspondences are only drawn from this candidate set henceforth. As we need a dense yet reliable set of correspondences, a modified sampling weight is used here.

Since the surfaces are quite well aligned in general after rigid alignment, we just need to rectify misalignments at a fine scale during non-rigid registration. These misalignments correspond to vertex pairs which are geometrically close but not yet consistent in color (*inconsistent* vertices). So instead of considering color distinctness of each vertex on its own surface, we now consider the color distinctness by taking both vertices in a pair, and considering both surfaces. To do so, for each vertex v^a on S^a, we find the nearest vertex v^p on S^p, again using reciprocal validation. Their Euclidean distance is put into an array, which is sorted in *ascending* order. Color differences based both on a single vertex (SingleColorDist) and on the mean value of the 1-ring of neighbors (MeanColorDist) are ranked in *descending* order: we use *mean* color distance here to improve robustness when judging whether a pair of vertices is a correct correspondence or not. Here $\text{MeanColorDist}(v_i, v_j) = \| M_{color}^i - M_{color}^j \|_2$, where M_{color}^i is defined in Eq. (1). After ranking, we retain a fraction $\eta = 0.75$ of vertices from each array. Vertices which are close in Euclidean distance,
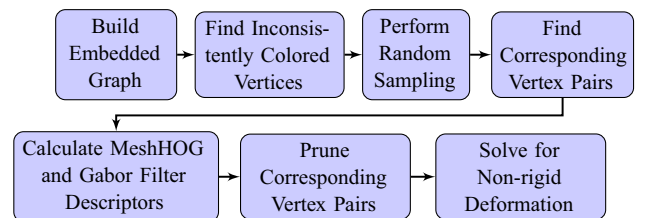

Fig. 4. Color-aware non-rigid registration pipeline.

but far in color, are marked as inconsistent. We now sample vertices from each surface, giving preference to inconsistent vertices by weights which control the probability of sampling. We use a weight of one for normal vertices, but a weight of N/N_I for inconsistent vertices, where N is the total number of vertices on the surface and N_I is the number of inconsistent vertices. Vertices are chosen randomly as in rigid registration.

After sampling, as shown in Algorithm 1, each sampling vertex is used to find multiple prospective matching vertices on the other surface which are filtered by reciprocal validation; remaining vertex pairs form candidate correspondences for the subsequent pruning step.

Algorithm 1. Find correspondences for color-aware non-rigid registration.

Input: Candidate vertex sets Ω^a and Ω^p, embedded graph E^a and E^p on each surface, kd-trees K^a and K^p built on S^a and S^p, number $K_{nonrigid}$ of closest vertices to consider, retention ratio κ used in metric ranking

Output: Corresponding vertex pair set P

1: **for** each vertex $v^a \in \Omega^a$ **do**
2: Set corresponding vertex set $C(v^a) = \varnothing$
3: Find $K_{nonrigid}$ closest vertices from K^p, put them into $C(v^a)$
4: **for** each vertex $v^p \in C(v^a)$ **do**
5:   **for** each graph node connected to v^a **do**
6:    Put SingleColorDist (v^a, v^p), MeanColorDist (v^a, v^p) and Euclidean distance between v^p and v^a into ranking arrays of current graph node
7:   **for**
8: **end**
9: **end**
10: **for** each graph node in E^a **do**
11: Rank its three metrics in ascending order and keep those vertex pairs all of whose metrics lie within the best fraction κ
12: **end**
13: **for** each retained vertex pair $\langle v^a, v^p \rangle$ **do**
14: If $\langle v^a, v^p \rangle$ wins local ranking of each related graph node to v^a, put v^p into Ω^p, put $\langle v^a, v^p \rangle$ in P^a.
15: **end**
16: Repeat process from Lines 1 to 9 for each vertex $v^p \in \Omega^p$ with K^a, perform local ranking from Lines 10 to 12 on E^p, and collect eligible vertex pair $\langle v^p, v^a \rangle$ as Lines 13 to 15 to build P^p.
17: **for** each vertex pair $\langle v^a, v^p \rangle \in P^a$ **do**
18: **if** $\langle v^p, v^a \rangle \in P^p$ **then**
19:   $P = P \cup \{\langle v^a, v^p \rangle\}$
20: **end**
21: **end**

3.2.2. Texture descriptors

MeshHOG descriptors are used in rigid registration. Being based on gradients, they are likely to be perturbed when the color or geometry is unreliable, particularly in challenging boundary areas or distorted areas. The descriptor only focuses on a small neighborhood and thus is unable to capture texture information well. To overcome this limitation, we generalize the Gabor filter descriptor [42] from 2D to mesh surfaces. Gabor filters are based on an integral method which captures textures at multiple scales and in different directions. Previous work has proved its effectiveness in 2D texture comparison.

In order to exploit Gabor filter descriptors on 3D textured surfaces, the Gabor filter is convolved with the texture projection at each vertex (see Fig. 5(b)). The area surrounding each vertex is
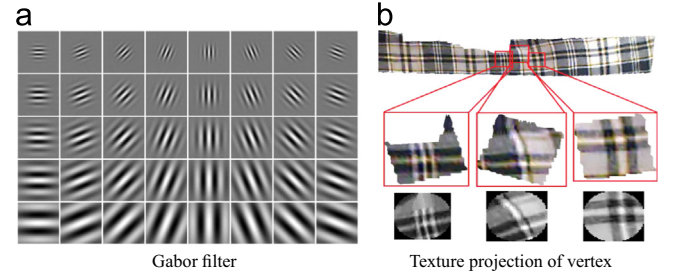
Fig. 5. Gabor filter and texture projected from the 3D surface for calculation of Gabor filter descriptor.

projected onto its tangent plane, and sampled to give a grayscale image. To obtain a rotation-invariant descriptor, we only keep pixels within a disk centered at each vertex. After convolution, to avoid the arbitrariness of tangent plane orientation, following [42], the output for different scales and orientations of the Gabor filter are circularly shifted so that the maximum filter output corresponds to the first orientation at the first scale, so that similar textures generate similar Gabor filter descriptors. The fast Fourier transform (FFT) is used for rapid convolution computation.

Fig. 6 shows some non-rigid registration results using either or both of these descriptors, revealing their different advantages. MeshHOG is more sensitive to sudden changes while Gabor filters are more robust in the presence of distortion e.g. in boundary areas: in Fig. 6(a), the pattern alignment in general is better, while in Fig. 6(b), texture alignment is better in distorted areas (which were blurred in the previous result). Combining them gives good alignment results everywhere: see Fig. 6(c).

### 3.2.3. Correspondence pruning using both color and geometric consistency

We notice that the distances between MeshHOG and Gabor descriptors of corresponding vertex pairs follow normal distributions. As MeshHOG and Gabor filters provide complementary information, we combine them to prune inconsistent correspondences jointly and use their weighted Gaussian errors to assess each remaining correspondence in Algorithm 2.

We first discard improbable pairs for which either MeshHOG or Gabor descriptor distances are larger than $\mu + \varepsilon\sigma$, where μ is the mean, σ is the standard deviation and $\varepsilon = 0.674$ to retain 75% of the distribution. The retained vertex pairs may still contain wrong correspondences, as correspondences have been considered in isolation. We can improve correspondences by considering their mutual consistency. We adapt the method in [43] which considers local isometric consistency and uses diffusion pruning. We first compute a matrix **M** measuring consistency of local geodesic distances between pairs of correspondences, then fill the diagonals of this matrix with measures which depend only on single correspondences rather than pairs. Unlike [43], the diagonal values are weighted probabilities based on distances of two texture descriptors, to reflect color texture consistency of correspondences. Intuitively, the probability should decrease with increasing descriptor distance; it should be normalized. We use the cumulative distribution functions C_{HOG} and C_{Gabor} as they are monotonic, in the range of $[0, 1]$ and can be explicitly obtained from the Gaussian distribution (see Algorithm 2). Each diagonal value is set to $W_{\langle v^a, v^p \rangle} = \alpha(1 - C_{HOG}) + (1 - \alpha)(1 - C_{Gabor})$. Diffusion pruning is then used to find a suitable set of correspondences.

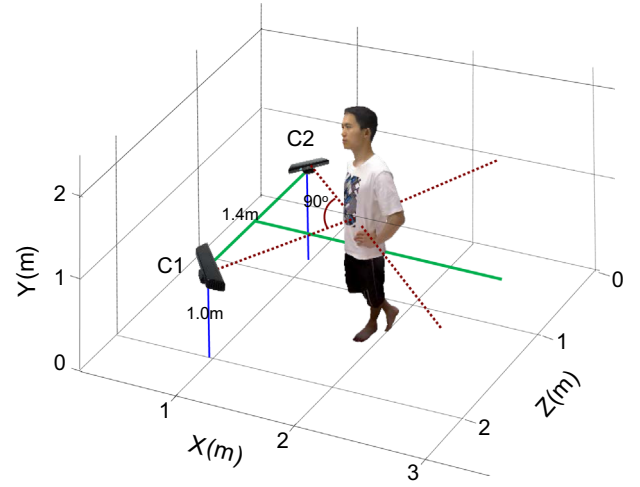**Algorithm 2.** Prune inconsistent corresponding vertex pairs.

Input: Corresponding vertex pair set P, MeshHOG descriptor sets Φ^a, Φ^p, Gabor filter descriptor sets Γ^a, Γ^p

Output: Final pruned corresponding vertex pair set P_f
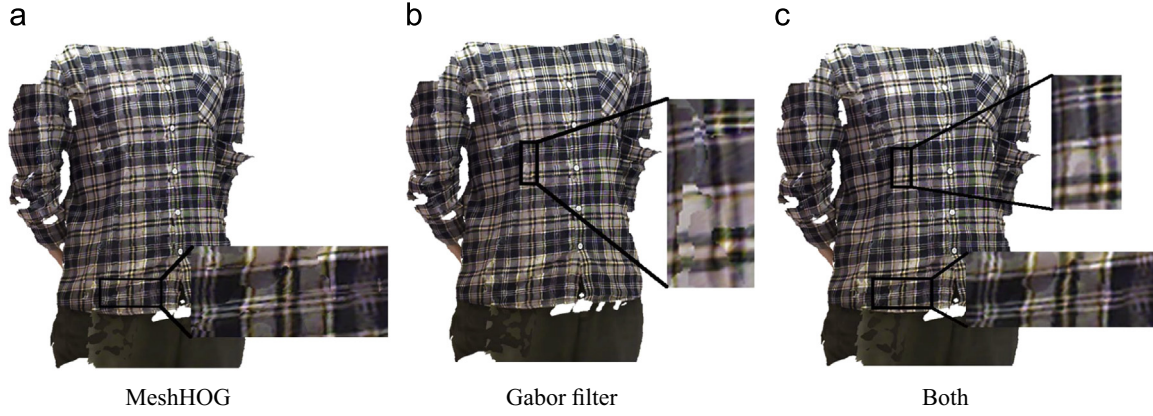
Fig. 6. Comparison of non-rigid registration results with one or both descriptors. (a) and (b) show two regions which cannot be handled well with a single descriptor, but are handled well after combining the descriptors together (c).

1: Compute the mean μ, standard deviation σ of MeshHOG and Gabor filter descriptors respectively, giving μ_{HOG}, σ_{HOG}, and μ_{Gabor}, σ_{Gabor}

2: **for** each vertex pair $\langle v^a, v^p \rangle$ in P **do**

3:    Compute MeshHOG and Gabor descriptor distances D_{HOG}, D_{Gabor}

4:    **if** $D_{HOG} > \epsilon \sigma_{HOG} + \mu_{HOG}$ **or** $D_{Gabor} > \epsilon \sigma_{Gabor} + \mu_{Gabor}$ **then**

5:       Delete $\langle v^a, v^p \rangle$ from P

6:    **else**

7:       Compute cumulative distribution function for D_{HOG} and D_{Gabor} as C_{HOG} and C_{Gabor}:

8:       $C_{HOG} = (1 + \text{erf}((D_{HOG} - \mu_{HOG})/(\sqrt{2}\sigma_{HOG})))/2$

9:       $C_{Gabor} = (1 + \text{erf}((D_{Gabor} - \mu_{Gabor})/(\sqrt{2}\sigma_{Gabor})))/2$

10:      Assign weight $W_{\langle v^a, v^p \rangle} = \alpha(1 - C_{HOG}) + (1 - \alpha)(1 - C_{Gabor})$

11:    **end**

12: **for**

13: Use local geodesic distance consistency for non-diagonals and weights $W_{\langle v^a, v^p \rangle}$ for diagonals and apply diffusion pruning to produce selected set P_f.

3.2.4. Non-rigid registration

Having found correspondences, an embedded-graph based approach is exploited to compute local deformations between the two surfaces, following [10]. We formulate the deformation by minimizing the energy in Eq. (3),

$$E = w_{rigid}E_{rigid} + w_{smooth}E_{smooth} + w_{corr}E_{corr}, \tag{3}$$

where the terms E_{rigid} and E_{smooth} measure the rigidity and smoothness of the embedded graph deformation respectively. E_{corr} measures the weighted Euclidean distance between corresponding vertex pairs:

$$E_{corr} = \sum W_{\langle v_i^a, v_j^p \rangle} \left\| v_i^a - v_j^p \right\|^2 \tag{4}$$

where $W_{\langle v_i^a, v_j^p \rangle}$ is defined previously. This weight gives a combined assessment of each correspondence based on its two descriptor differences. Our implementation sets $w_{rigid} = 10$, $w_{smooth} = 2$, $w_{corr} = 10$. Fig. 12 shows corresponding alignment results.

4. Results

In this section, we first briefly describe the experimental setup and parameter settings, then evaluate our algorithm and compare it to other state-of-the-art registration techniques.

Fig. 7. Physical configuration of two calibrated Kinect cameras.

4.1. Textured surfaces captured by two calibrated Kinects

Human data was captured using two calibrated Kinect cameras to generate textured 3D surfaces. The cameras were located on the left and right sides of the subject as shown in Fig. 7. Their orientations were at 90° to each other; they were about 1.4 m apart, and 1.0 m above the floor. The subject is about 1.0 m from the line between the Kinects, along its perpendicular bisector. The two Kinects were calibrated using the method in [44], giving the 3D coordinates of each 3D point captured by the two Kinects in the same reference frame. The two Kinects capture the data successively: first, the left camera turns on and captures a single RGB-depth frame, then turns off, then the right Kinect does likewise. This avoids interference between the Kinects: attempting to *simultaneously* capture two depth maps results in major quality degradation in overlapping areas due to the technology used. The time between capture of these two frames is about 1 s, during which time, a person always moves slightly, even if trying to hold a static pose. In the raw data captured by the Kinect, each frame is a partial scan of the captured subject, comprising a 640×480 color image registered with a 640×480 depth image. The two captured RGB-D images are converted to a 3D mesh surface with texture information using the OpenNI package.

4.2. Parameter settings

While our method involves several parameters, they were fixed in all our experiments. For rigid registration they were set to: K_{rigid}

=20 and κ=0.5. Parameters in Algorithms 1 and 2 for non-rigid registration were set to: $K_{nonrigid}$ = 50, κ=0.5, α=0.5. Clearly, the retention ratio parameter κ used in searching for correspondences and size of the embedded graph nodes are the two key parameters in our approach. We thus first experimentally justify our choices for these parameters.

4.2.1. Retention ratio κ

The retention ratio κ is in [0, 1]. We considered values from 0.1 to 0.8 in steps of 0.1, and tested registration performance for the same pair of surfaces under various starting conditions, with increasing difficulty of registration for cases from 1 to 6 in Fig. 8. The final registration result is shown in the bottom row of Fig. 11. For each pair of surfaces, we used different values of κ to register them 50 times and collected statistics.

Fig. 9 compares the average number of iterations, the number of corresponding vertices, and their average color distance, after registration using varying κ. Smaller κ removes bad correspondences more easily, which speeds up convergence and provides better registration accuracy. However, larger κ avoids local minima and is more robust to different starting conditions. As we may assume that the initial transformation is quite well known from the camera configuration, for this experimental setup, we can choose a smaller value of κ. Taking speed of convergence, final registration accuracy and robustness into account, κ=0.5 is a good compromise.

4.2.2. Number of embedded graph nodes

During non-rigid registration, the embedded graph is used not only to describe surface deformation, but also to organize local ranking when searching for correspondences, so the size and distribution of graph nodes has an impact on results. Our algorithm distributes the graph nodes evenly on the surface (as is usual): doing so is both easy and effective. Fig. 10 considers six different choices for the number of graph nodes. Each bar shown represents a specific ratio of number of graph nodes to the total number vertices on the surface. Different fractions of embedded graph nodes were tested on the surfaces shown in Fig. 1(a). We considered two statistics: the ratio of inconsistent vertices before and after registration, and the ratio of distance moved, which is a normalized value which represents the average change in position change of each active vertex due to registration.

If graph nodes are too many, each node considers vertex pairs from too local a perspective, so local alignment may be trapped in

a local minimum and robustness cannot be guaranteed; vertices' positions tend to oscillate rather than converging to the correct position. If graph nodes are too few, local ranking reduces to global ranking, so only correspondences with global consistency will be retained. Using fewer graph nodes means we cannot handle fine details of surface deformation so well, and we are more likely to find a global near-rigid transformation: non-rigid registration does little to improve the initial rigid registration. As a compromise, we use 0.5% of the number of surface vertices as the number of embedded graph nodes.

4.3. Evaluation

We now evaluate our method for effectiveness and robustness, and then consider limitations. Although our method involves random sampling, our tests show that we use sufficient samples to ensure almost identical results in repeated runs.

4.3.1. Effectiveness

The aim of our technique is to produce *visually high-quality* registered textured surfaces for cases where color misalignment is particularly noticeable. We thus demonstrate the effectiveness of our method via visual comparisons with the results of other rigid [4,7,21] and non-rigid [24] registration methods, using several datasets.

Registration results from five pairs of scans are shown in Figs. 11 and 12. The data in Examples 1–4 were captured using two calibrated Kinects. The surfaces involve slight non-rigid deformation; they were aligned by our rigid registration algorithm first. To ensure a fair comparison with other non-rigid registration
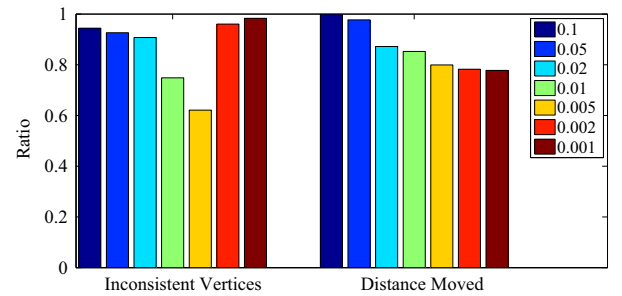

Fig. 10. Registration results for different numbers of embedded graph nodes. The legend gives the numerical ratio of graph nodes to surface vertices.
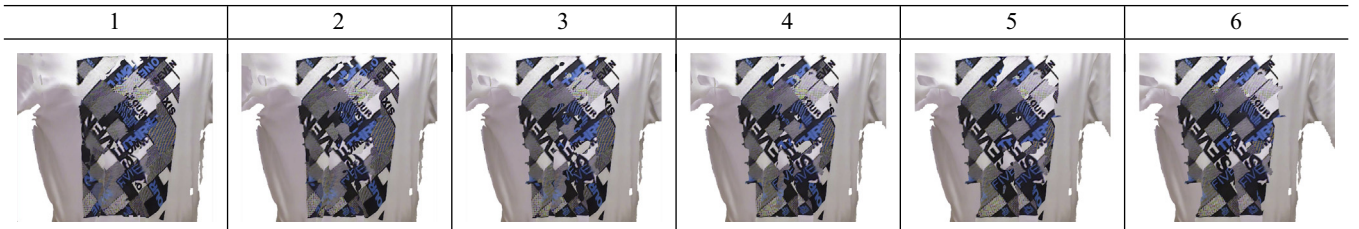

Fig. 8. Different starting conditions for determining the best value of κ for rigid registration. See the last row of Fig. 11 for aligned results.
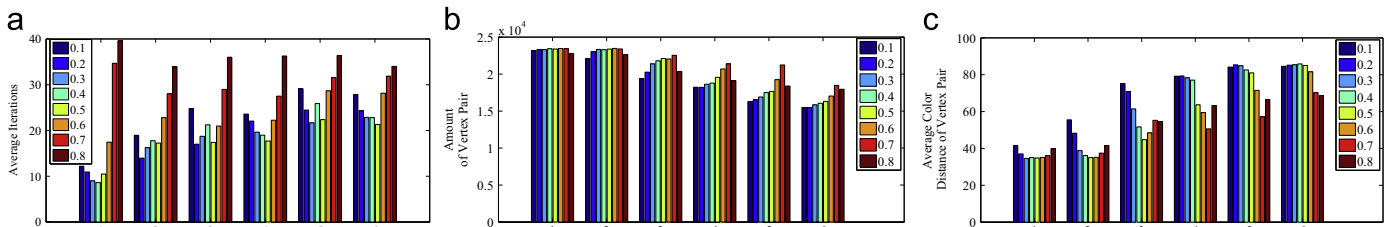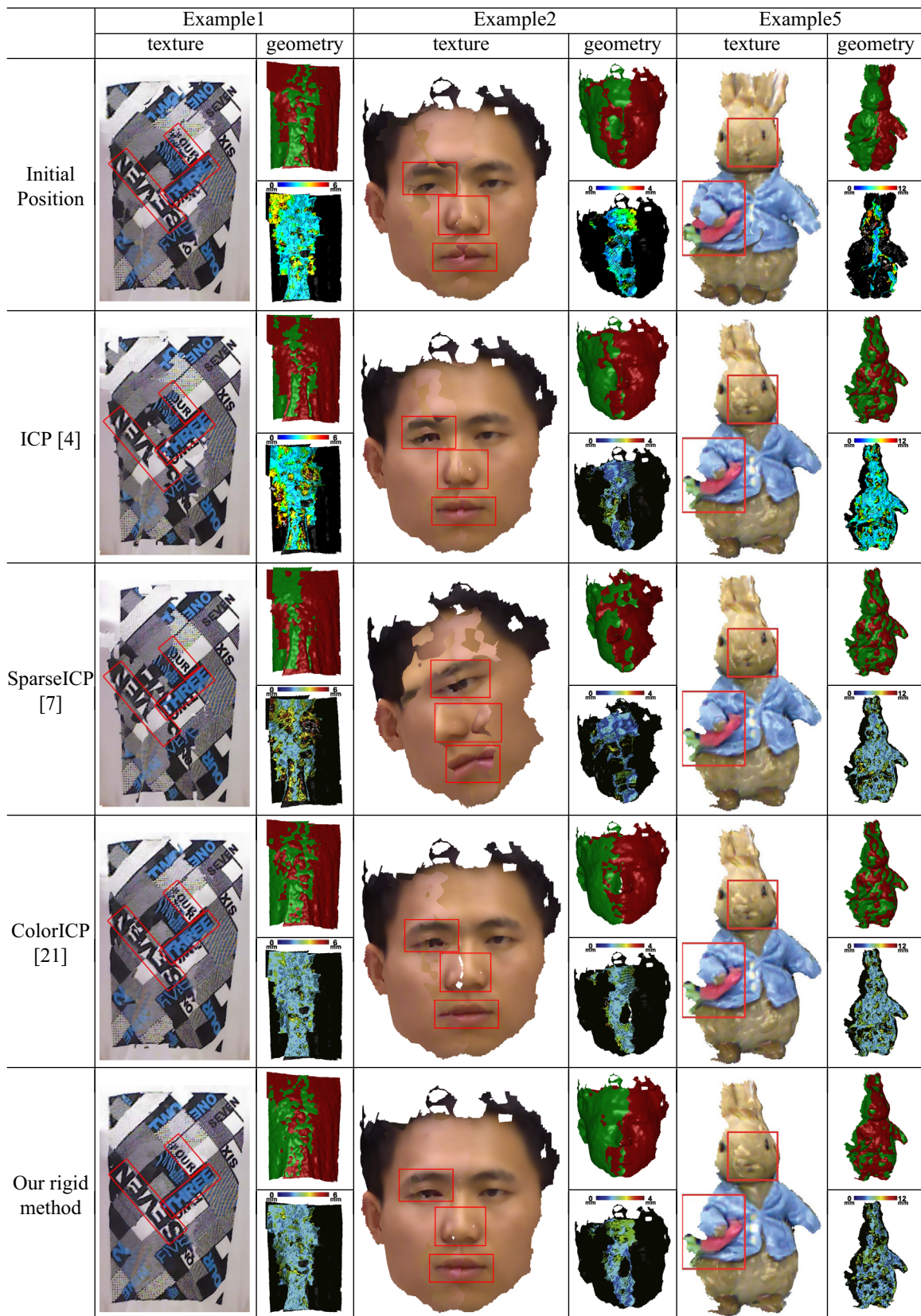

Fig. 9. Registration results for varying retention ratio κ: average number of iterations, number of corresponding vertices, and their average color distance.

Fig. 11. Comparison of our rigid registration method to other state-of-the-art algorithms. Alignment differences in texture can be clearly seen in regions marked by red rectangles on the left of each example. On the right, geometry corrections are shown in registration results without texture; pseudo-color rendering of closest vertex pair distances before and after registration: the distance ranges are 0–6 mm, 0–4 mm and 0–12 mm for the three examples. Vertices for which a closest vertex pair could not be found are rendered as black (largely for non-overlapping regions), with distance thresholds the same as the maximum of the above range in each case. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)
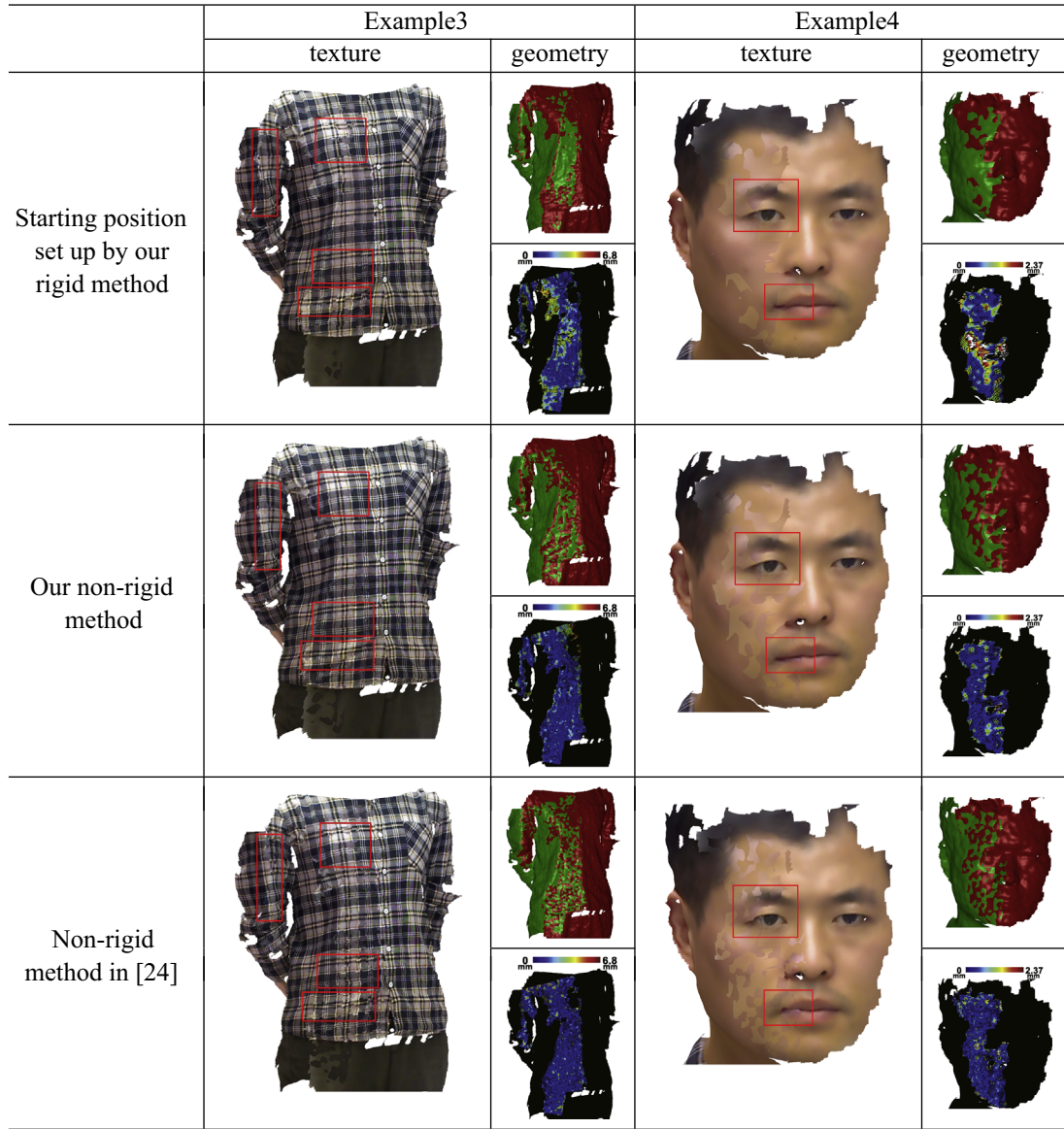
Fig. 12. Different alignment effects produced by our non-rigid registration method and the method in [24]. The latter is able to correct geometric discrepancies, as shown by the untextured registration results in which pseuodocolor gives closest vertex pair distances before and after registration. For the two examples, the distance ranges are 0–6.8 mm and 0–2.37 mm. Vertices for which a closest vertex pair could not be found are rendered as black (largely for non-overlapping regions), with distance thresholds the same as the maximum of the above range in each case. Our method is better at rectifying texture misalignments, especially in the regions marked by red rectangles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 1
Mean distance of closest vertex pair after rigid alignment for examples in Fig. 11 (mm).

	Initial	ICP [4]	SparseICP [7]	ColorICP [21]	Our rigid method
Example 1	2.56	1.86	2.15	1.50	1.25
Example 2	1.67	1.30	1.36	0.93	1.06
Example 5	2.38	1.55	1.44	1.34	1.30

methods, the same rigid registration output was used as initialization in each case. Example 5 used a dataset from [17], and only involves rigid transformation; nonrigid registration is not used. Parameter settings for various methods were those given by their authors. We use Examples 1, 2, 5 for rigid registration result comparisons and Examples 3, 4 for non-rigid comparisons. Similar conclusions apply to all examples.

Our color-aware rigid registration method produces better results than the other methods, as shown in Fig. 11, thanks to the effective use of color texture information in the registration framework. See in particular the registered regions in red rectangles: our texture visual alignment is much more accurate than that produced by other methods [4,7,21]. ICP [4] and sparseICP [7] do not use color information. In ColorICP [21], color information is directly incorporated into the ICP framework.

Geometric registration results are also shown in Fig. 11, using pseudo-color rendering of closest vertex pair distances for overlapping regions and black for points for which no suitable correspondences were found (mainly in non-overlapping regions). The mean distance of closest vertex pairs in overlapping regions is also provided in Table 1, both the initial values, and values after registration by various methods [4,7,21] including ours. Each method converged monotonically to a minimum—the result was smaller than the initial value in each case. However, the method in [7] converged to an incorrect local minimum for Example 2, with misalignment of the nose.

RGB-D cameras often capture higher resolution color images than depth images. Color provides valuable additional information, allowing our registration framework to produce better results than purely geometric methods. Our color descriptors were further designed to be robust to illumination differences between scans, and relative distortions. By using color descriptors, our method makes better use of color than ColorICP [21], as Fig. 11 shows.

Our non-rigid registration further improves texture alignment. The only previous non-rigid registration method using color [24] uses color descriptors with a consistency constraint between successive frames, assuming continuous data capture. They base non-rigid registration on a pairwise method [10], constrained by color descriptors. Therefore, starting from the same initial rigid alignment, we compare non-rigid registration results with [24] via
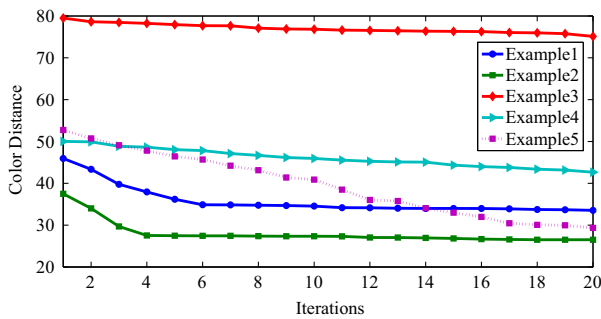
pair-wise registration. Our results in Fig. 12 have better visual quality. We use pseudocolor in overlapping regions to visualize geometric distances between the vertices of the registered surfaces in Fig. 12.

4.3.2. Convergence

The convergence of our method is demonstrated by registration results for Examples 1–5 in Figs. 11 and 12; corresponding statistics are provided in Figs. 13 and 14.

For rigid registration, Fig. 13 shows the average color distances for vertex pairs in the overlapping region. During iteration, as

Table 2
Run times for rigid/non-rigid registration steps (ms).

	Prepare	Sampling	Descriptor	Search	Prune	Postprocess
Rigid	359	2777	59	160		28
Non-rigid	3494	1242	56	14,632	58,902	780

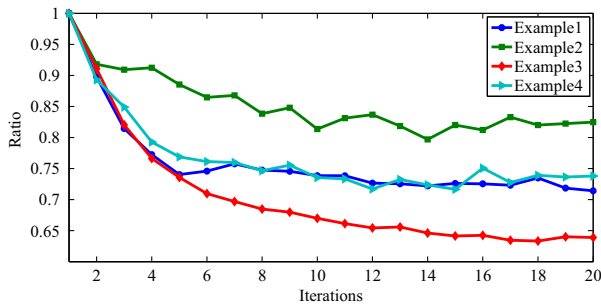**Fig. 13.** Mean color distance in rigid registration with respect to number of iterations.

Fig. 14. Fraction of vertices remaining inconsistent during non-rigid registration with respect to the number of iterations.
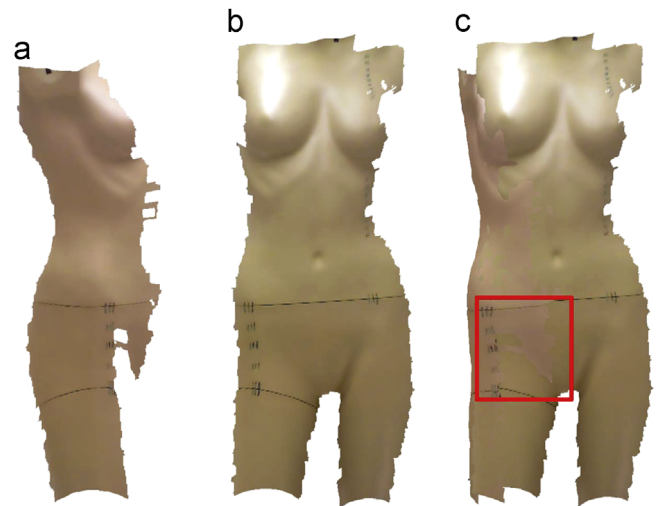

Fig. 16. Little-textured surfaces. (a) and (b) are parts captured from a mannequin model with pure color. There is little texture in the red rectangle. It is too difficult to generate sufficient vertex pairs depending on color/texture information, so the registration is prone to geometric discrepancies (c). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Fig. 15. Robustness in mid-scale registration. (a) The initial partial data before registration. Minor pose and cloth wrinkle changes cause difficulties especially if we try to use non-rigid alignment. (b) Overlap of the partial data before and after registration. (c) Euclidean distance changes for the closest vertex pairs before and after registration. Distances in the range 0–15 mm are rendered in pseudo-color. Vertices for which a closest vertex pair could not be found (largely for non-overlapping regions) are rendered as black, with a distance threshold of 15 mm.

more and more of these vertices become correctly aligned, the average color distance reduces, gradually converging to a fixed residual. For Example 3, surfaces need to be warped to eliminate misalignment, so rigid registration cannot improve it beyond a certain limit, and after that, further decrease in color distance is minor.

Using non-rigid registration, the inconsistent vertices (vertices geometrically close to the matched vertices but with large color deviation) give a good indication of convergence. Fig. 14 shows the fraction of remaining inconsistent vertices compared to the initial number: our algorithm improves color consistency by iterative non-rigid alignment.

4.3.3. Robustness

In order to test the robustness of our method to larger than expected non-rigid deformations, we chose two similar surfaces captured from the same torso in different poses, as shown in Fig. 15(a). Our algorithm successfully registers the parts with vertices with same texture. Fig. 15(c) show the change of Euclidean distance for the same vertex pairs before and after registration. These distances reduce after registration, indicating the robustness of our algorithm when presented with mid-scale deformation.

In general, it is difficult to ensure that objects are evenly illuminated and all images are equally bright: surfaces captured by different cameras show differences in overall intensity. This is noticeable in all the examples in this paper. Our algorithm handles this issue well, and is robust to a certain degree of illumination change.

4.3.4. Timing

As we utilize two kinds of color descriptors as well as a pruning strategy, our algorithm takes longer than algorithms which do not use color. Table 2 gives run times of our algorithm per iteration, for two surfaces each containing 95K vertices. These statistics were obtained on a computer with a 1.6 GHz Intel Core i7 processor and 4 GB memory. The *Prepare* stage covers preprocessing including building kD-trees and loading descriptors from file, while the *Postprocess* stage concerns assembling and solving equations, as well as transforming surfaces. For speed, we precompute the descriptors and used a hash table to store and reuse them. Our method takes 3 s and 76 s per iteration respectively for rigid and non-rigid cases.

4.4. Limitations

Our method has certain limitations. We assume that coarse alignment is available with reasonable amounts of overlap between scans. This is acceptable when the camera configuration can be controlled, but may be more difficult to achieve in general settings such as a hand-held capture system. In such cases, coarse registration techniques based on e.g. intrinsic properties could be used to initialize our method. Even with precomputation, the runtime is quite high, and further acceleration e.g. by GPU, should be investigated. Our method achieves improved registration by exploiting color information. For surfaces with little texture, our method degrades to traditional rigid or non-rigid registration using only geometric information. See Fig. 16 for an example.

5. Conclusions

We have proposed a novel algorithm to register 3D surfaces with detailed textures. It is distinctive in use of two color descriptors to improve the finding of corresponding vertex pairs. Color is also taken into account when computing the transformation (in the rigid case) or deformation needed to bring the surfaces into alignment (in the non-rigid case). In both cases, our experiments show that our method can accurately align surfaces with detailed textures, outperforming earlier approaches. The descriptors we currently use are based on either color texture or geometry. In the future, we hope to directly combine color and intrinsic geometry which could potentially provide a more effective joint descriptor for registration. Our current approach registers two scans. Applying our method sequentially to multiple scans may not produce ideal results as registration errors can accumulate. We wish in future to extend the method to multiple scans using ideas such as group registration.

References

[1] Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm. In: 3DIM; 2001. p. 145–52.

[2] Pottmann H, Huang QX, Yang YL, Hu SM. Geometry and convergence analysis of algorithms for registration of 3D shapes. Int J Comput Vis 2006;67(3):277–96.

[3] Tam G, Cheng ZQ, Lai YK, Langbein F, Liu Y, Marshall D, et al. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. IEEE Trans Vis Comput Graph 2013;19:1199–217.

[4] Besl PJ, McKay ND. A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 1992;14:239–56.

[5] Chen Y, Medioni G. Object modelling by registration of multiple range images. Image Vis Comput 1992;10:145–55.

[6] Phillips J, Liu R, Tomasi C. Outlier robust ICP for minimizing fractional RMSD. In: 3-D Digital imaging and modeling; 2007. p. 427–34.

[7] Bouaziz S, Tagliasacchi A, Pauly M. Sparse iterative closest point. In: Symposium on geometric processing; 2013. p. 113–23.

[8] Yang J, Li H, Jia Y. Go-ICP: solving 3D registration efficiently and globally optimally. In: ICCV; 2013. p. 1457–64.

[9] Albarelli A, Rodolà E, Torsello A. Fast and accurate surface alignment through an isometry-enforcing game. Pattern Recogn 2015;48(7):2209–26.

[10] Li H, Sumner RW, Pauly M. Global correspondence optimization for non-rigid registration of depth scans. In: Symposium in geometric processing; 2008. p. 1421–30.

[11] Wand M, Adams B, Ovsjanikov M, Berner A, Bokeloh M, Jenke P, et al. Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data. ACM Trans Graph 2009;28 15:1–15.

[12] Zheng Q, Sharf A, Tagliasacchi A, Chen B, Zhang H, Sheffer A, et al. Consensus skeleton for non-rigid space-time registration. Comput Graph Forum 2010;29:635–44.

[13] Yang J, Li K, Li K, Lai YK. Sparse non-rigid registration of 3D shapes. Comput Graph Forum 2015;34(5):89–99.

[14] Johnson A, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans Pattern Anal Mach Intell 1999:433–49.

[15] Krsek P, Pajdla T, Hlaváč V, Martin R. Range image registration driven by a hierarchy of surface differential features. In: Workshop of Austrian association for pattern recognition; 1998, p. 175–83.

[16] Gal R, Cohen-Or D. Salient geometric features for partial shape matching and similarity. ACM Trans Graph 2006;25(1):130–50.

[17] Tombari F, Salti S, Di Stefano L. Unique signatures of histograms for local surface description. In: ECCV; 2010. p. 356–69.

[18] Bokeloh M, Berner A, Wand M, Seidel HP. Schilling, A. Slippage features. Technical report; 2008.

[19] Johnson AE, Kang SB. Registration and integration of textured 3-D data. In: 3-D Digital imaging and modeling; 1997. p. 234–41.

[20] Druon S, Aldon M, Crosnier A. Color constrained ICP for registration of large unstructured 3D color data sets. In: Information acquisition; 2006, p. 249–55.

[21] García-Bermejo JG, Casanova EZ, Alonso RF. Automated registration of 3D scans using geometric features and normalized color data. Comput-Aided Civil Infrastruct Eng 2013;28:98–111.

[22] Brusco N, Andreetto M, Giorgi A, Cortelazzo G. 3D registration by textured spin-images. In: 3-D Digital imaging and modeling; 2005. p. 262–9.

[23] Holowko E, Wojsz J, Sitnik R, Karaszewski M. Color-based algorithm for automatic merging of multiview 3D point clouds. J Comput Cult Herit 2014;7 16:1–21.

[24] Li H, Luo L, Vlasic D, Peers P, Popović J, Pauly M, et al. Temporally coherent completion of dynamic shapes. ACM Trans Graph 2012:2.

[25] Zhou QY, Koltun V. Color map optimization for 3D reconstruction with consumer depth cameras. ACM Trans Graph 2014;33(4):155.

[26] Bronstein M, Kokkinos I. Scale-invariant heat kernel signatures for non-rigid shape recognition. In: CVPR; 2010. p. 1704–11.

[27] Sfikas K, Pratikakis I, Theoharis T. ConTopo: Non-rigid 3D object retrieval using topological information guided by conformal factors. In: 3D Object retrieval workshop; 2011. p. 25–32.

[28] Darom T, Keller Y. Scale-invariant features for 3-D mesh models. IEEE Trans Image Process 2012;21(5):2758–69.

[29] Aubry M, Schlickewei U, Cremers D. The wave kernel signature: a quantum mechanical approach to shape analysis. In: ICCV workshops; 2011. p. 1626–33.

[30] Zaharescu A, Boyer E, Varanasi K, Horaud R. Surface feature detection and description with applications to mesh matching. In: CVPR; 2009. p. 373–80.

[31] Aganj E, Monasse P, Keriven R. Multi-view texturing of imprecise mesh. In: ACCV; 2010. p. 468–76.

[32] Dellepiane M, Marroquim R, Callieri M, Cignoni P, Scopigno R. Flow-based local optimization for image-to-geometry projection. IEEE Trans Vis Comput Graph 2012:463–74.

[33] Gal R, Wexler Y, Ofek E, Hoppe H, Cohen-Or D. Seamless montage for texturing models. Comput Graph Forum 2010:479–86.

[34] Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P. et al. Kinect-Fusion: real-time 3D reconstruction and interaction using a moving depth camera. In: ACM symposium on user interface software and technology; 2011. p. 559–68.

[35] Tong J, Zhou J, Liu L, Pan Z, Yan H. Scanning 3D full human bodies using Kinects. IEEE Trans Vis Comput Graph 2012;18(4):643–50.

[36] Li H, Vouga E, Gudym A, Luo L, Barron JT, Gusev G. 3D self-portraits. ACM Trans Graph 2013;32(6):187.

[37] Bouaziz S, Tagliasacchi A, Pauly M, Dynamic 2D/3D registration. In: Eurographics technical course notes; 2014.

[38] Zollhöfer M, Nießner M, Izadi S, Rehmann C, Zach C, Fisher M, et al. Real-time non-rigid reconstruction using an RGB-D camera. ACM Trans Graph 2014;33 (4) 156:1–12.

[39] Innmann M, Zollhöfer M, Nießner M, Theobalt C, Stamminger M. Volume-Deform: real-time volumetric non-rigid reconstruction. Technical report arxiv:1603.08161; 2016.

[40] Huhle B, Magnusson M, Strasser W, Lilienthal AJ. Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform. In: International conference on robotics and automation; 2008. p. 4025–30.

[41] Endres F, Hess J, Engelhard N, Sturm J, Cremers D, Burgard W. An evaluation of the RGB-DSLAM system. In: International conference on robotics and automation; 2012. p. 1691–6.

[42] Rahmana M, Pickering M, Frater M. Scale and rotation invariant Gabor features for texture retrieval. In: Digital image computing tech appl; 2011. p. 602–7.

[43] Tam GKL, Martin RR, Rosin PL, Lai YK. Diffusion pruning for rapidly and robustly selecting global correspondences using local isometry. ACM Trans Graph 2014;33(1):4.

[44] Chen Y, Dang G, Cheng ZQ, Xu K. Fast capture of personalized avatar using two kinects. J Manuf Syst 2014;33(1):233–40.