

Projective Surface Matching of Colored 3D Scans

Kari Pulli

Nokia Research Center & MIT

kari.pulli@nokia.com

Simo Piironen

Nokia

simo.piironen@nokia.com

Tom Duchamp

Univ. Washington, Seattle

duchamp@math.washington.edu

Werner Stuetzle

Univ. Washington, Seattle

wxs@stat.washington.edu

Abstract

We present a new method for registering multiple 3D scans of a colored object. Each scan is regarded as a color and range image of the object recorded by a pinhole camera. Consider a pair of cameras that see overlapping parts of the objects. For correct camera poses, the actual image of the overlap area in one camera matches the rendition of the overlap area as seen by the other camera. We define a mismatch score summarizing discrepancies in color, range, and silhouette between pairs of images, and we present an algorithm to efficiently minimize this mismatch score over camera poses.

1. Introduction

Merging parts of a surface scanned from different viewpoints by transforming them into a common coordinate system (*registration*) is a fundamental part of 3D scanning. We want to choose registration transformations so that the regions of the surface seen in multiple scans are well-aligned—they should be geometrically close and their colors should match.

Incorporating color information into the registration process is important. First, geometry alone may not sufficiently constrain the transformations. For instance, if the overlap between scans is flat, then the surface pieces can slide past each other without penalty. However, color patterns may provide sufficient information to uniquely determine the registration. Second, merging color information is necessary for creating consistent texture maps for the reconstructed surface. Misregistrations that might be barely visible in the geometry may lead to unacceptable blurring or ghosting of color features.

1.1. Previous work

Previous approaches for aligning colored range scans [2, 9, 16, 20] usually solve this problem by a variant of the Iterated Closest Point (ICP) method [3]. First, using a heuristic, points are paired, and then using the paired points

a rigid 3D transformation that best aligns the pairs is calculated. The hope is that now the scans move into better registration, the points that are matched next are more likely to correspond to the same object surface point, a better alignment can be computed, and the iterated process converges to a good registration. This approach has been generalized to work with several views instead of just two [15].

The original ICP method used the 3D distance as the correspondence heuristic, pairing a point with the closest point in the other data set. Several enhancements to this heuristic have been proposed. The color can be used as an extra constraint such that points can only be paired with nearby points of similar color [9]. Another variant is to include the color difference or the difference in the local normal vector orientation as part of the distance metric [8]. Combining various heuristics allows quite fast implementations [17].

Another approach to solve the correspondence problem is to project the data sets onto the image plane of one of the range sensors. The simplest approach is to pair points projecting to the same location [4, 13]. If color or intensity data is available, it can be used to improve matches, e.g., by using the intensity gradients and differences to find a better match on the projection plane, and back-projecting to get a 3D point pair [20]. Taking this idea forward, one can try to do a more thorough matching on the projected color data by using 2D image registration to apply a planar projective warp that aligns the color images and then pairing the points [16], or by searching the neighborhood and pairing locations that maximize the cross-correlations within a small window [2]. In all these methods the matched 3D points are passed to the alignment stage that finds the rigid 3D motion that best aligns the matched point pairs.

The empty-space constraint (no part of another well-registered scan can lie in front of the visible surface or outside its silhouette) has been used in surface modeling (e.g., [6]). Lensch *et al.* [11] use the silhouette information exclusively when registering color images to a 3D model that has accurate geometry but no color information. They extract the silhouette from a color image, project the 3D model using the camera model of the color camera, and match the projected silhouette with the extracted one.

Also related to our work are methods that use projection to reconstruct range data from stereo image pairs [7, 10].

1.2. Projective Surface Matching

Previous approaches to registration using both color and range data amount to solving an alternating sequence of two different optimization problems. In the first optimization step, color and geometric information are used to find point correspondences; in the second step, the registration transformation is then determined using point correspondences alone. However, there is no guarantee that this procedure will converge because the procedure does not optimize a well defined objective function.

In our approach, which we call Projective Surface Matching (PSM), we assess closeness of geometry and color by projecting the data onto the image plane of an imaginary pinhole camera. With perfect data and perfect registration, points in different scans corresponding to the same surface location project to the same location on the image plane. However, there is no need for checking the match on every imaginary camera plane; associating a camera with each scan is sufficient.

We define a mismatch score summarizing discrepancies in color, range, and silhouette between images of different surface pieces, and we present an algorithm to efficiently minimize this mismatch score over camera poses.

In addition to the range and color information, we use an empty space constraint penalizing registrations for which registered data project outside of the actually observed silhouettes.

For simplicity of exposition we restrict ourselves to the case of two scans. However, our approach generalizes to registration of multiple scans.

1.3. Outline of the paper

In the next section we set up the geometric description of the registration problem, define a mismatch score measuring registration error, and present an algorithm for minimizing the mismatch score. In Section 3 we describe implementation details of our method, such as hierarchical processing to avoid local minima and choice of relative weights for the components of the mismatch score. Section 4 presents experimental results. The remaining sections contain a discussion and a summary of the method. In an appendix we derive formulas for the gradients of various functions required by the optimization algorithm.

2. Description of Projective Surface Matching

We model a scanner by a pinhole camera with focal length f , center of projection \mathbf{o} , and an orthonormal frame

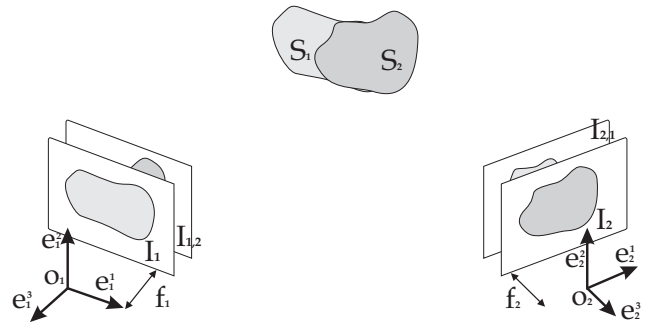


Figure 1. Two scans and their virtual cameras.

$\mathbf{e} = (\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3)$, where \mathbf{e}^1 and \mathbf{e}^2 span the image plane and \mathbf{e}^3 points away from the scene. For a given pose (\mathbf{o}, \mathbf{e}) , a camera maps a point $\mathbf{x} \in \mathbb{R}^3$ in its field of view into a point $\mathbf{u} = (u, v) = \Pi(\mathbf{x})$ on its image plane:

$$\Pi(\mathbf{x}) = f \left(\frac{(\mathbf{x} - \mathbf{o}) \cdot \mathbf{e}^1}{(\mathbf{x} - \mathbf{o}) \cdot \mathbf{e}^3}, \frac{(\mathbf{x} - \mathbf{o}) \cdot \mathbf{e}^2}{(\mathbf{x} - \mathbf{o}) \cdot \mathbf{e}^3} \right). \quad (1)$$

Imaging a colored surface \mathbf{S} gives rise to a range function $r(\mathbf{u}) = (\mathbf{x} - \mathbf{o}) \cdot \mathbf{e}^3$ and a color function $\mathbf{c}(\mathbf{u})$ defined on $\Omega = \Pi(\mathbf{S})$. On the other hand, an image (Ω, r, \mathbf{c}) and a pose together define a (part of a) colored surface in \mathbb{R}^3 . A point $\mathbf{u} \in \Omega$ corresponds to a point $\mathbf{x} = \mathbf{P}(\mathbf{u}) \in \mathbb{R}^3$:

$$\mathbf{P}(\mathbf{u}) = \mathbf{o} + \frac{r(\mathbf{u})}{f} (u \mathbf{e}^1 + v \mathbf{e}^2 + f \mathbf{e}^3). \quad (2)$$

The color at $\mathbf{x} = \mathbf{P}(\mathbf{u})$ is $\mathbf{c}(\mathbf{u})$.

2.1. Definition of the mismatch score

Suppose that we have imaged \mathbf{S} with two cameras as illustrated in Figure 1, but we do not know their poses. We assume that (i) the field of view of each camera contains the entire object, and (ii) the parts of \mathbf{S} seen by the two cameras overlap. Let \mathcal{I}_1 be the image recorded by camera 1. Our goal is to estimate the pose of camera 2. Given an initial guess $(\mathbf{o}_2, \mathbf{e}_2)$ for the pose of camera 2, we can convert the image \mathcal{I}_2 into a colored surface \mathbf{S}_2 .

Mapping \mathbf{S}_2 into camera 1 produces an image $\mathcal{I}_{1,2} = (\bar{\Omega}, \bar{r}, \bar{\mathbf{c}})$. Note that since \mathbf{S}_2 is only a partial view of the surface \mathbf{S} , some parts of \mathbf{S}_2 must be occluded from camera 1. Correctly dealing with occlusions is critical; we return to this point in Section 2.3.

If the guess for the pose of camera 2 is correct, then $\bar{\Omega} \subset \Omega$ and $r(\mathbf{u}) = \bar{r}(\mathbf{u})$, $\mathbf{c}(\mathbf{u}) = \bar{\mathbf{c}}(\mathbf{u})$ for all $\mathbf{u} \in \Omega \cap \bar{\Omega}$. If the guess is not correct, we want to change the pose or, equivalently, apply a Euclidian transformation $\mathbf{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to \mathbf{S}_2 . Mapping $\mathbf{T}(\mathbf{S}_2)$ into camera 1 gives a new image $\mathcal{I}_{1,2}(\mathbf{T}) = (\bar{\Omega}_T, \bar{r}_T, \bar{\mathbf{c}}_T)$. We define a loss function measur-

ing the mismatch between the images \mathcal{I}_1 and $\mathcal{I}_{1,2}(\mathbf{T})$:

$$L(\mathbf{T}) = \int_{\Omega \cap \bar{\Omega}_T} ((r - \bar{r}_T)^2 + \kappa_1 \|\mathbf{c} - \bar{\mathbf{c}}_T\|^2) \quad (3)$$

$$+ \kappa_2 \int_{\bar{\Omega}_T} d^2(\mathbf{u}, \Omega).$$

The first term measures mismatch of the range and color functions; the last term measures the extent to which the domain $\bar{\Omega}_T$ of $\mathcal{I}_{1,2}$ extends beyond the domain Ω of \mathcal{I}_1 . Here $d(\mathbf{u}, \Omega)$ is the distance between a point \mathbf{u} and its closest point in Ω .

Reversing the roles of cameras 1 and 2, we can analogously define a loss function $\bar{L}(\mathbf{T}^{-1})$. Our final mismatch score is $M(\mathbf{T}) = L(\mathbf{T}) + \bar{L}(\mathbf{T}^{-1})$. Our goal is to find the transformation $\hat{\mathbf{T}} = \operatorname{argmin}_T M(\mathbf{T})$ minimizing the mismatch score.

2.2. Minimizing the mismatch score

Minimizing $M(\mathbf{T})$ is a nonlinear optimization problem which we solve by an iterative algorithm. We first find a small Euclidian transformation \mathbf{T}_1 close to the identity \mathbf{I} with $M(\mathbf{T}_1) < M(\mathbf{I})$. We then update the poses and repeat the process until M fails to decrease. Updating the poses ensures that at any step of the algorithm we are only searching for a transformation near the identity. After n iterations our estimate of the optimal transformation is $\mathbf{T}_n \circ \mathbf{T}_{n-1} \circ \dots \circ \mathbf{T}_1$.

One approach to reducing the mismatch is gradient descent: compute a finite difference approximation \mathbf{G} to $\nabla_T M(\mathbf{T})$ at $\mathbf{T} = \mathbf{I}$ and then search for a step size s minimizing $M(\mathbf{I} - s\mathbf{G})$. With three rotation and three translation parameters, a robust finite difference approximation requires 12 evaluations; moreover, gradient descent tends to be slow.

We have used an alternative approach, which is hard to rigorously justify but is much more efficient.

Consider for the moment the term $L(\mathbf{T})$; $\bar{L}(\mathbf{T}^{-1})$ can be treated analogously. Define the *optical flow* $\psi_T : \bar{\Omega} \rightarrow \bar{\Omega}_T$ by

$$\psi_T(\mathbf{u}) = \Pi \circ \mathbf{T} \circ \bar{\mathbf{P}}(\mathbf{u}). \quad (4)$$

Optical flow translates movement of a point on \mathbf{S}_2 under the transformation \mathbf{T} into movement on the image plane. Using the definition of optical flow we can rewrite the second component of $L(\mathbf{T})$ (Eqn. (3)). Applying a change of variables $\mathbf{u} = \psi_T(\mathbf{v})$ yields the approximation

$$\int_{\bar{\Omega}_T} d^2(\mathbf{u}, \Omega) d\mathbf{u} \approx \int_{\bar{\Omega}} d^2(\psi_T(\mathbf{v}), \Omega) d\mathbf{v} \quad (5)$$

where the domain of integration no longer depends on \mathbf{T} . We neglect the Jacobian $|\frac{\partial \mathbf{u}}{\partial \mathbf{v}}|$ which is close to 1 for $\mathbf{T} \approx \mathbf{I}$.

Replacing the integrals in Eqn. (3) by sums over pixels, the approximation to $L(\mathbf{T})$ becomes

$$L^*(\mathbf{T}) = \sum_{\mathbf{u}_i \in \Omega \cap \bar{\Omega}_T} ((r - \bar{r}_T)^2 + \kappa_1 \|\mathbf{c} - \bar{\mathbf{c}}_T\|^2)$$

$$+ \kappa_2 \sum_{\mathbf{u}_i \in \bar{\Omega}} d^2(\psi_T(\mathbf{u}_i), \Omega). \quad (6)$$

Except for the dependence of the first summation domain on the parameter \mathbf{T} , $L^*(\mathbf{T})$ looks like the objective function of a nonlinear least squares problem — it is of the form $\sum_i f_i^2(\mathbf{T})$ where the f_i are nonlinear functions of \mathbf{T} . The approximation $\bar{L}^*(\mathbf{T}^{-1})$ to the other component of the mismatch score can be expressed in a similar way. Therefore, for some functions f_i and g_j ,

$$M(\mathbf{T}) \approx M^*(\mathbf{T}) = \sum_i f_i^2(\mathbf{T}) + \sum_j g_j^2(\mathbf{T}^{-1}).$$

We use the Levenberg-Marquardt method [14] to find an incremental transformation at $\mathbf{T} = \mathbf{I}$ that reduces $M^*(\mathbf{T})$. Strictly speaking, the way in which Levenberg-Marquardt generates its search directions is only justified if the summation domains do not depend on the parameter \mathbf{T} . In practice, however, this does not seem to matter here.

To apply Levenberg-Marquardt we need the gradients of the functions $f_i(\mathbf{T})$ and $g_j(\mathbf{T}^{-1})$ at $\mathbf{T} = \mathbf{I}$. Notice that

$$\nabla_T g_i(\mathbf{T}^{-1}) = -\nabla_T g_i(\mathbf{T})$$

when $\mathbf{T} = \mathbf{I}$. Examination of the formulas for f_i and g_j shows that their gradients can be expressed in terms of the gradients of \bar{r}_T , $\bar{\mathbf{c}}_T$, and ψ_T , which we derive in the appendix.

2.3. Dealing with occlusions

The occlusion problem is illustrated in Figure 2. It shows an object (colored in grey) seen by two cameras, labeled \mathbf{c}_1 and \mathbf{c}_2 . Cameras 1 and 2 see the red and blue parts of the object's surface, respectively. Notice that points a and c are visible in both cameras (and can therefore be considered both red and blue), while point b is only visible in camera 2 and d is only visible in camera 1. Mapping the blue parts of the surface into camera 1 correctly identifies the red a with the blue a —they project onto the same point in the image plane. However, matching of points based on their images in camera 1 also erroneously identifies the blue point b with the red point d . The problem is that camera 2 cannot see the part of the red surface containing d that occludes the point b from camera 1.

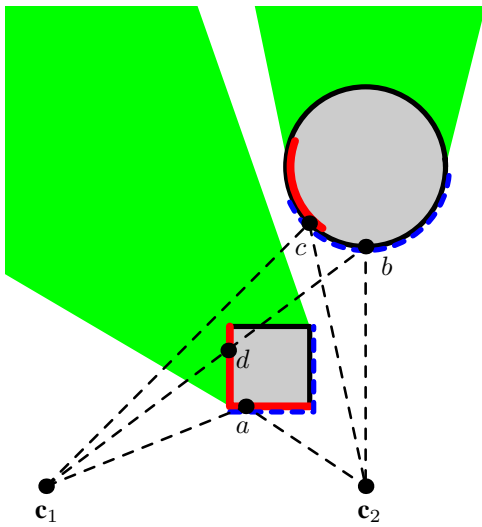


Figure 2. Extruding silhouettes prevents two different surfaces from projecting to the same pixel.

One approach to eliminating false matches is to identify points on the red and blue surfaces projecting onto the same location in the image plane only if their distances differ by no more than a given threshold [2, 16]. We have taken an alternative approach that does not require thresholds. It is inspired by *shadow volumes* [5]. If camera 2 is treated as the light source, the shadow volume is the interior of the region defined by the visible (blue) surface and its extruded silhouette. The shadow volume is the union of the green and grey areas in Figure 2.

Based on the information available from camera 2, the object could be any region in the shadow volume whose boundary contains the blue surface. To get a conservative criterion for deciding which of the blue points are visible to camera 1, we assume that the object fills the entire shadow volume. Note that in Figure 2 the extruded silhouette occludes the point b from camera 1, preventing a mismatch. Note also that the approach is conservative; even though point c is actually visible in both cameras, it is flagged as occluded.

We implement the occlusion calculation by rendering the extruded silhouettes in a color that we do not expect to find on the object.

3. Implementation

In this section we describe various implementation details.

3.1. Projecting the data

We convert the range scans into triangle meshes which are texture mapped with the color data.

A virtual pinhole camera is associated with each scan using the scanner’s parameters if those are available. Otherwise, we fit a camera model directly to the data; range maps give a dense mapping from 3D points to 2D image points, and we can estimate the pose of the pinhole camera that best reproduces that mapping [19].

Each scan is then projected to the image plane of its virtual camera by rendering it using the OpenGL graphics library. For each pixel we obtain the color and range by reading the color and z-buffers. We calculate image plane gradients using finite differences (a 3×3 Sobel operator). For each pixel outside the silhouette we find the closest silhouette pixel.

3.2. Hierarchical processing

We perform the optimization using a hierarchy of scales, starting with a small image size (low resolution), optimizing while the mismatch score increases rapidly and then moving to a larger image size until the original resolution is reached. This hierarchical approach saves time: processing smaller images is quicker. In addition, the hierarchical approach decreases the likelihood of getting stuck in a local minimum.

3.3. Scaling the different components

It is important to balance the translation and rotation components of \mathbf{T} so that they have similar effect on the optical flow. First we have to choose the location of the pivot of rotation \mathbf{m} . The best place is at the intersection of surface normals, e.g., for a spherical object that would be the center of the sphere. In practice we take the center of mass of the scan, and since that often is very close to the surface, we displace it further behind the data. Second, we scale the scan so that on average the data are at unit distance from \mathbf{m} . That way, unit changes in either angle or translation have commensurate effects [18]. Color values need to be scaled so that they have the same magnitude as range values.

The weight κ_2 of the silhouette error should be set so that it drives the scans behind the silhouettes. However, it should not be so large that color and range errors cannot be minimized because an intermediate step would move the scans so that they project slightly beyond the silhouettes. We scale the silhouette error so that at the distance of one third of the image width the error is one unit. That is, if the image width is w , $\kappa_2 = (w/3)^{-2}$.

Since range and color data are usually unreliable near silhouettes, we attenuate the pixel contribution by a smooth

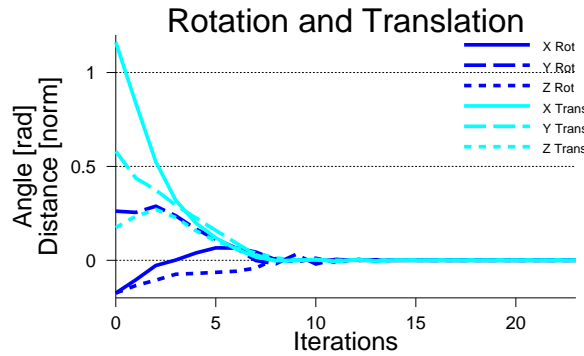
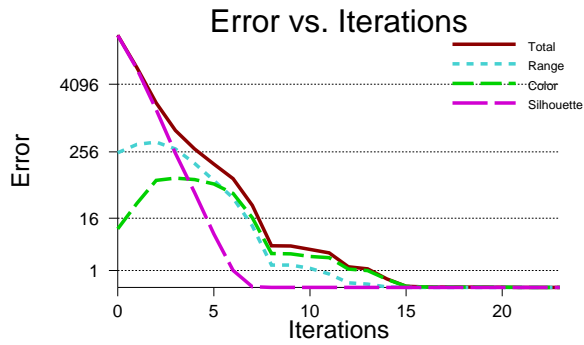


Figure 3. Registering a scan against itself.

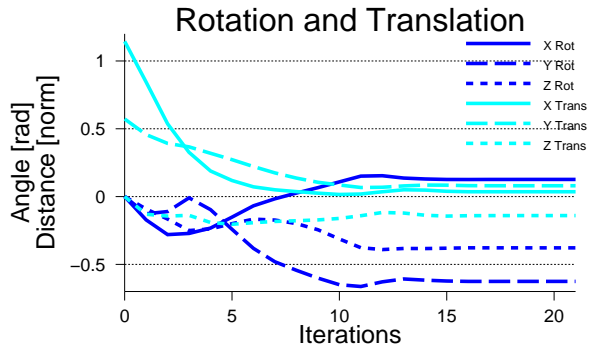
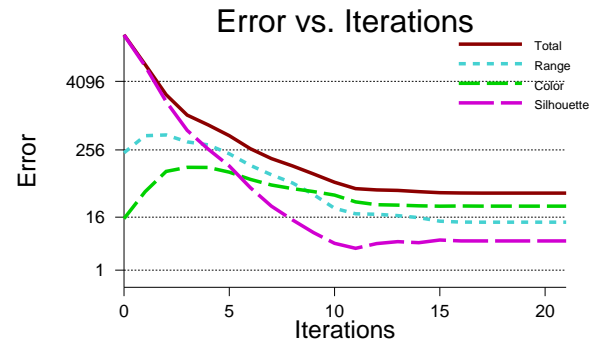


Figure 4. Progress of the registration.

weighting function that is zero at the boundary, quickly approaches one inside the boundary, and grows more gradually to one outside the boundary.

4. Results

As a sanity check, we registered a scan of a toy rabbit against a displaced copy of itself. Figure 3 shows the progress of the registration starting from a poor guess for initial registration. The progress is fast, and a good registration is obtained.

We tested our method with scans from various sources. Figure 4 shows the errors and transformations parameters when registering the two scans in Figure 5. The top row in Figure 5 shows \mathcal{I}_1 and \mathcal{I}_2 , while the bottom row shows $\mathcal{I}_{1,2}$ and $\mathcal{I}_{2,1}$. We see that, initially, \mathcal{S}_2 projects mostly outside Ω_1 , and \mathcal{S}_1 projects mostly outside Ω_2 . The cyan “shadow” in $\mathcal{I}_{1,2}$ and $\mathcal{I}_{2,1}$ is the occlusion volume. Recall that the occlusion volume is like a shadow cone that covers the parts of the scan that we want to exclude from the error evaluation. In the error graph (Figure 4), we see that at first the silhouette error dominates, and, as a larger percentage of the scan projects within the silhouette, the color and range errors increase though the total error decreases.

By iteration 4 the scans have moved so that they mostly

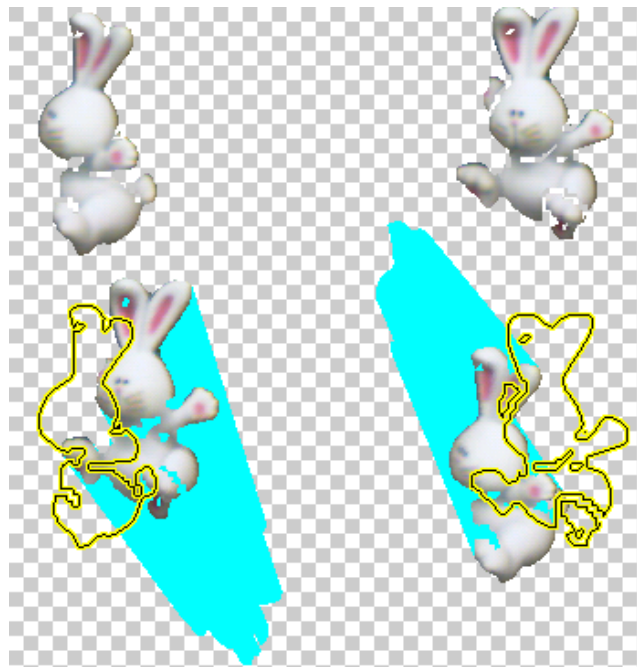


Figure 5. The starting configuration. Top-down, left-right: \mathcal{I}_1 , \mathcal{I}_2 , $\mathcal{I}_{1,2}$, $\mathcal{I}_{2,1}$.

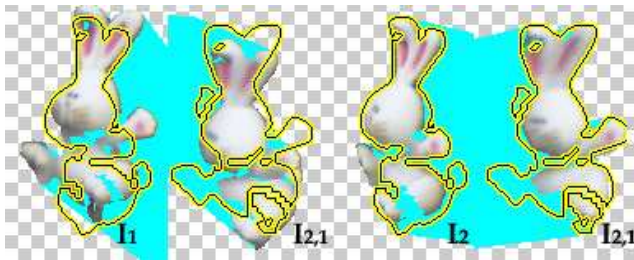


Figure 6. Intermediate (left) and final (right) configuration.

fall within the silhouettes, but are misaligned. This situation is illustrated on the left side of Figure 6. In the next few iterations the scans are rotated to improve the fit. By the 10th iteration, the registration is good and the rotations and translation do not change by much in subsequent iterations. Color and range errors diminish until the algorithm achieves convergence at around iteration 15. The final configuration is shown on the right side of Figure 6. Some error remains as the scans are not perfect in geometry (range error), color (e.g., slightly different illumination conditions), or silhouette (scanner did not scan all visible surfaces, for example the right leg of the bunny was missing from the left scan).

To demonstrate how using color information can help where geometry alone fails we created synthetic scans of an undulating cylindrical vase that we texture mapped with a color image. The upper block in Figure 7 shows the registration results from the two cameras (first row) as well as the error images (second row) when all three of the error terms (range, color, silhouette) were used. The registration is quite good, though some color error remains. This is due to the different sampling of color and geometry in the two scans, which is an inherent problem with any scanner, real or simulated. The bottom block shows the result of using only the range and silhouette error terms. The geometry aligns well, but there is significant misalignment in the color data.

5. Discussion

5.1. Initial registration

In general, nonlinear optimization methods require a reasonable initial guess for the parameters. However, we have observed that our method is robust and can cope with large initial registration errors. For the most part, this is due to the empty-space constraint. Another reason is our hierarchical coarse-to-fine approach. In some examples (e.g., Figure 8), optimization at the fine level of detail does not converge to



Figure 7. Registering with color works (top two rows; color data and differences), without color fails. Different samplings of the same data still produce some color error after registration (second row).

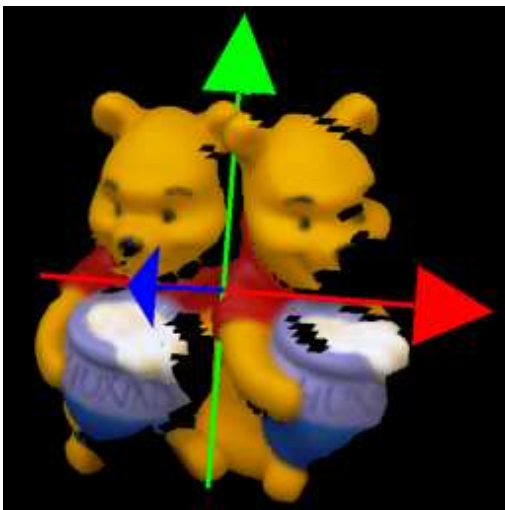


Figure 8. At this starting configuration only the hierarchical approach finds the correct solution.

a good solution, while the hierarchical optimization does. Initial errors in rotation angles pose greater problem than errors in translation parameters. Nevertheless, errors up to 30 degrees can still be consistently tolerated, and in Figure 8 the initial rotation error about the y-axis was 40 degrees.

5.2. Changes in the lighting

The simple color comparison that we used in Section 2 works well when the illumination is constant over scans and the surface of the object is diffuse. If these assumptions are violated, more sophisticated color processing is required. Matching colors under different illumination is a non-trivial problem. Simplistic approaches, such as using HVS color space (hue, value, saturation) and ignoring the value component or dividing out the luminance of RGB, do not work reliably. With the former, colors close to gray give essentially a random hue, while the latter has similar problems with all dark colors. If the lighting is calibrated, one can try to compensate for the effects of direct illumination and extract the diffuse reflectance [12]. Another possibility is to use normalized cross-correlation of colors, which is more robust to illumination changes than the direct color value [1].

5.3. Use of normal vectors

Our method is not limited to using only range and color data. We can incorporate any other function on the surface, such as the normal vector field to the surface. Figure 9 illustrates why using normals can improve registration. The figure shows two scans that are close to each other, so that

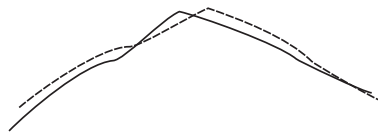


Figure 9. Pieces of scans are close to each other, but the corners are not aligned.

range data would not necessarily align the corners. However, the large differences between normal vectors close to the corners might help to pull the scans into a better alignment.

One way to include normal vectors into our algorithm would be to encode them as texture maps. We include the analysis of normal vectors in the appendix.

6. Summary

We have presented a registration method that measures registration accuracy by projecting range scans, along with associated color information, onto the image planes of virtual cameras. While previous methods use color information to determine correspondences between surface points, we minimize a single error function measuring misalignment in both color and range.

The use of the empty-space constraint (silhouette penalty), the use of the occlusion volume to prevent false matches, and the hierarchical coarse-to-fine processing, make the algorithm robust against errors in the initial registration.

Acknowledgements

The range scan repository of Patrick Flynn at Ohio State University was a good source of data for our experiments. We'd also like to thank Keijo Ruotsalainen for his help in formulating some of the equations.

Appendix

Our goal is to derive the gradients of color $\bar{\mathbf{c}}_T$, range \bar{r}_T , normal $\bar{\mathbf{n}}_T$, and optical flow ψ_T at $\mathbf{T} = \mathbf{I}$. Ignoring the effects of occlusion and disocclusion, we can express $\bar{\mathbf{c}}_T$, \bar{r}_T , and $\bar{\mathbf{n}}_T$ in terms of the optical flow:

$$\bar{\mathbf{c}}_T(\mathbf{u}) = \bar{\mathbf{c}} \circ \psi_{T^{-1}}(\mathbf{u}), \quad (7)$$

$$\bar{r}_T(\mathbf{u}) = \mathbf{e}^3 \cdot (\mathbf{T} \circ \mathbf{P} \circ \psi_{T^{-1}}(\mathbf{u})). \quad (8)$$

$$\bar{\mathbf{n}}_T(\mathbf{u}) = \mathbf{R} \cdot \bar{\mathbf{n}} \circ \psi_{T^{-1}}(\mathbf{u}). \quad (9)$$

A Euclidean transformation \mathbf{T} for a point $\mathbf{x} = (x, y, z)$ is composed of a rotation \mathbf{R} around some pivot $\mathbf{m} =$

(m_x, m_y, m_z) followed by a translation \mathbf{t} :

$$\mathbf{T}(\mathbf{x}) = \mathbf{R} \cdot (\mathbf{x} - \mathbf{m}) + \mathbf{m} + \mathbf{t}. \quad (10)$$

For the normal \mathbf{n} it is simply

$$\mathbf{T}(\mathbf{n}) = \mathbf{R} \cdot \mathbf{n}. \quad (11)$$

We parameterize Euclidean transformations by the 6-dimensional vector $\mathbf{d} = (\alpha, \beta, \gamma, t_1, t_2, t_3)$ of rotation angles around the three coordinate axes and translations.

Using the chain rule we get

$$\nabla_{\mathbf{d}} \bar{\mathbf{c}}_{\mathbf{d}} = -\nabla_{\mathbf{u}} \bar{\mathbf{c}} \cdot \nabla_{\mathbf{d}} \psi_{\mathbf{d}}, \quad (12)$$

$$\nabla_{\mathbf{d}} \bar{\mathbf{r}}_{\mathbf{d}} = -\nabla_{\mathbf{u}} \bar{\mathbf{r}} \cdot \nabla_{\mathbf{d}} \psi_{\mathbf{d}} + \mathbf{e}^3 \cdot (\nabla_{\mathbf{d}} \mathbf{T} \circ \bar{\mathbf{P}}), \quad (13)$$

$$\nabla_{\mathbf{d}} \bar{\mathbf{n}}_{\mathbf{d}} = -\nabla_{\mathbf{u}} \bar{\mathbf{n}} \cdot \nabla_{\mathbf{d}} \psi_{\mathbf{d}} + \nabla_{\mathbf{d}} \mathbf{R} \cdot \bar{\mathbf{n}}. \quad (14)$$

Recall that we evaluate all gradients at $\mathbf{T} = \mathbf{I}$, i.e., at $\mathbf{d} = \mathbf{0}$ and that $\nabla_{\mathbf{d}} \psi_{\mathbf{T}^{-1}} = -\nabla_{\mathbf{d}} \psi_{\mathbf{T}}$ (again at $\mathbf{T} = \mathbf{I}$).

The gradients $\nabla_{\mathbf{u}} \bar{\mathbf{c}}$, $\nabla_{\mathbf{u}} \bar{\mathbf{r}}$, and $\nabla_{\mathbf{d}} \bar{\mathbf{n}}_{\mathbf{d}}$ are the derivatives of the data with respect to image coordinates and are estimated numerically.

By the chain rule, $\nabla_{\mathbf{d}} \psi = \nabla_{\mathbf{x}} \Pi \cdot \nabla_{\mathbf{d}} (\mathbf{T} \circ \bar{\mathbf{P}})$, where

$$\nabla_{\mathbf{x}} \Pi = \frac{1}{r(\mathbf{u})} \begin{bmatrix} f\mathbf{e}^1 - u\mathbf{e}^3 \\ f\mathbf{e}^2 - v\mathbf{e}^3 \end{bmatrix} \quad (15)$$

and

$$\nabla_{\mathbf{d}} \mathbf{T}(\mathbf{x}) = \begin{bmatrix} 0 & z - m_z & -(y - m_y) & 1 & 0 & 0 \\ -(z - m_z) & 0 & (x - m_x) & 0 & 1 & 0 \\ (y - m_y) & -(x - m_x) & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (16)$$

Hence, writing $\mathbf{P}(\mathbf{u}) = \mathbf{x}$ yields the formula

$$\nabla_{\mathbf{d}} \psi = \frac{1}{r(\mathbf{u})} \begin{bmatrix} f\mathbf{e}^1 - u\mathbf{e}^3 \\ f\mathbf{e}^2 - v\mathbf{e}^3 \end{bmatrix} \cdot \begin{bmatrix} 0 & z - m_z & -(y - m_y) & 1 & 0 & 0 \\ -(z - m_z) & 0 & (x - m_x) & 0 & 1 & 0 \\ (y - m_y) & -(x - m_x) & 0 & 0 & 0 & 1 \end{bmatrix}.$$

References

- [1] P. Beardsley, P. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. In *Proc. European Conference on Computer Vision*, pages 683–695, 1996.
- [2] F. Bernardini, I.M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Trans. Visualization and Computer Graphics*, 7(4):318–332, 2001.
- [3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(2):239–256, February 1992.
- [4] G. Blais and M. D. Levine. Registering multiview range data to create 3d computer objects. *IEEE Trans. Patt. Anal. Machine Intell.*, 17(8):820–824, August 1995.
- [5] F. Crow. Shadow algorithms for computer graphics. In *SIGGRAPH 77 Conference Proceedings*, Annual Conference Series, pages 242–248. ACM SIGGRAPH, 1977.
- [6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH 96 Conference Proceedings*, pages 303–312. ACM SIGGRAPH, Addison Wesley, August 1996.
- [7] P. Fua. From multiple stereo views to multiple 3-d surfaces. *Int. Journal of Computer Vision*, 24(1):19–35, August 1997.
- [8] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3-d model building. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 581–586, 1994.
- [9] G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Proc. SPIE vol. 2350: Videometrics III*, pages 279–290, 1994.
- [10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.
- [11] H. Lensch, W. Heidrich, and H-P. Seidel. A silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63:245–262, May 2001.
- [12] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo project: 3D scanning of large statues. In *Computer Graphics, SIGGRAPH 2000 Proceedings*, pages 131–144, 2000.
- [13] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, May 1995.
- [14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.
- [15] K. Pulli. Multiview registration for large data sets. In *Proc. 3-D Digital Imaging and Modeling*, pages 160–168, October 1999.
- [16] K. Pulli and L. G. Shapiro. Surface reconstruction and display from range and color data. *Graphical Models and Image Processing*, 62(3):165–201, 2000.
- [17] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. 3-D Digital Imaging and Modeling*, pages 145–152, May 2001.
- [18] D. A. Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Carnegie Mellon University, 1996.
- [19] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.
- [20] S. Weik. Registration of 3-d partial surface models using luminance and depth information. In *Proc. 3-D Digital Imaging and Modeling*, pages 93–100, May 1997.