

Fast 3D Model Acquisition from Stereo Images

Louis-Philippe Morency Ali Rahimi Trevor Darrell

MIT AI Lab

Cambridge, MA 02139

{lmorency,rahimi,trevor}@ai.mit.edu

Abstract

We propose a fast 3D model acquisition system that aligns intensity and depth images, and reconstructs a textured 3D mesh. 3D views are registered with shape alignment based on intensity gradient constraints and a global registration algorithm. We reconstruct the 3D model using a new Cubic Ray Projection merging algorithm which takes advantage of a novel data structure: the linked voxel space. Finally, we present experiments to test the accuracy of our approach on 3D face modeling using real-time stereo images.

1. Introduction

We present a technique for building textured 3D models which stitches together synchronized range and intensity frames from stereo cameras. Range from real-time stereo provides the basis for a modeling tool that is small and hand-held, requires one-time only calibration, is completely passive, produces almost instant 3D models, and provides real-time feedback as the model is being acquired.

The acquisition process is divided in two steps, common to most reconstruction algorithms [4]: 1) registering the frames to recover their relative positions in the real world and 2) reconstructing a 3D model by merging these frames. A registration step is necessary because the shape of most objects cannot be observed from only one view: we must scan the object from several directions and bring these scans into registration. But because frames can rarely be brought into exact registration, a merging phase is required to resolve these conflicts by forcing points to lie on a 2D manifold.

Using correlation-based stereo yields significantly noisier range information than traditional range scanners, requiring model acquisition methods that take advantage of intensity information for alignment. Our gradient-based registration algorithm employs an efficient global registration technique that allows it to take into consideration all frames in the sequence simultaneously, improving registra-

tion significantly. The outcome of the registration phase is a 3D mesh transformed to a canonical pose where each vertex correspond to a valid image pixel.

Due to noise in the imager and imperfect registration, the vertices will not lie on a 2D manifold, but will instead form a fuzzy cloud around the desired surface. The Cubic Ray Projection algorithm non-rigidly deforms each mesh so that vertices are forced toward a 2D manifold. To facilitate the creation of connected meshes from unstructured range data, we use a linked voxel space during the merging process. The linked voxel space is easily turned into a connected mesh for rendering.

The system presented is extremely fast. Many 3D views are merged together, reducing noise in the final model. When used with a real-time stereo camera, it is possible to capture 3D models interactively and unobtrusively.

1.1. Previous Work

Many algorithms have been proposed for registering range data. These differ notably in the energy function minimized during registration, and whether the registration procedure ensures global consistency.

The method of Stoddart and Hilton[16] minimizes a function corresponding to the energy stored in a spring system connecting corresponding points across frames. This algorithm provides global consistency, but requires correspondences to be known.

The registration algorithm of [6] brings each point of a scan as close as possible to its closest point on the model acquired so far, thus avoiding the need for correspondences. However, since this method does not produce a globally consistent model, as accumulated registration errors against the model eventually cause the model to become inconsistent (see [15] for a discussion).

The Iterated Closest Point (ICP) framework proposed by Besl and McKay [1] iteratively assigns correspondences and then minimizes the resulting distance metric by rigidly transforming the scans [12, 3]. Chen and Medioni [3] employ this technique to minimize the distance between each point of a scan and the closest tangent plane in the corre-

sponding scan. They perform this minimization jointly over the pose of all scans. Because each iteration must involve all pairs of corresponding points, the optimization is expensive.

To reduce the complexity of this minimization, Pulli [12] first aligns scans pairwise, obtaining relative pose estimates between many redundant pairs of scans. Global consistency is obtained by assigning each frame a pose such that the pairwise relative alignments are minimally perturbed. This optimization is fast as it does not require correspondences to be recomputed at each iteration of the optimization and only matches up frame poses instead of individual points.

Our approach uses the combined depth and intensity constraint of [7] to obtain relative pose changes between each frame and several other base frames. The pose changes describe the rigid transformation required for bringing each frame into registration with its base frames. The global registration method we present is based on [13] and is similar in structure to [12] in that, during global registration, poses are relaxed to find a registration which is consistent with the measured pairwise pose changes.

The following sections reviews the pose change estimation algorithm of [7] and the global pose consistency algorithm of [13]. Section 3 describes our novel 3D model reconstruction algorithm called Cubic Ray Projection, which is applied after frames have been globally registered. We then show how our system can be used to build 3D models of human heads.

2. Registration

To recover the motion between two frames, we apply the traditional Brightness Change Constraint Equation (BCCE) [9] jointly with the Depth Change Constraint Equation (DCCE) of [7] on range and intensity imagery of stereo camera. The BCCE finds motion parameters which minimize the appearance difference between the two frames in a least-squares sense:

$$\begin{aligned} \delta^* &= \arg \min_{\delta} \epsilon_{BCCE}(\delta) \\ \epsilon_{BCCE} &= \sum_x \|I_t(x) - I_{t+1}(x + u(x; \delta))\|^2 \quad (1) \end{aligned}$$

where $u(x; \delta)$ is the image flow at pixel x , parameterized by the details of a particular motion model. In the case of 3D rigid motion under a perspective camera, the image flow becomes:

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} (\delta_{\omega} \times X + \delta_{\Delta}), \quad (2)$$

where X is the world coordinate of the image point x , δ_{ω} is the infinitesimal rotation of the object, δ_{Δ} is its infinitesimal translation, and f is the focal length of the camera[2].

The DCCE of [7] uses the same functional form as equation (1) to constrain changes in depth. But since depth is not preserved under rotation, the DCCE includes an adjustment term:

$$\epsilon_{DCCE} = \sum_x \|Z_t(x) - Z_{t+1}(x + u(x; \delta)) + V_z(x; \delta)\|^2,$$

where V_z is the flow towards the Z direction induced by δ . Note that the DCCE is robust to lighting changes since lighting does not affect the depth map. We combine the BCCE and DCCE into one function optimization function with a weighted sum:

$$\delta^* = \arg \min_{\delta} \epsilon_{BCCE}(\delta) + \lambda \epsilon_{DCCE}(\delta),$$

The only unknown variables are the pose parameters, since Z is available from the depth maps. For an approximate way to optimize this function, see [7], where one iteration of Newton-Raphson is shown to be adequate for tracking.

2.1. Global Registration

Given a method for computing the pose difference δ_s^t between frames I_s and I_t , one approach for estimating the pose ξ_t of frame I_t relative to the first frame I_0 is to accumulate the pose difference between adjacent frames I_s and I_{s+1} , for $s = 0..t-1$. But since each pose change measurement is noisy, the accumulation of these measurements becomes noisier with time, resulting in unbounded drift.

To curb this drift, we compute the pose change between I_t and several base frames. When the trajectory of the target crosses itself, its pose change is computed against recently acquired scans as well as past scans near the current pose. These pose differences are combined to not only obtain a more robust and drift-free pose estimate of the current scan, but also to adjust the pose of past frames by incorporating knowledge about the closed trajectory.

Several authors have proposed an optimization framework to implement this technique [14, 12, 11]. Poses are assigned to each scan so that the predicted pose changes between pairs of scans are as similar as possible to the observed pose changes. Assuming a function $d(\xi_s, \xi_t)$ which returns the pose change between two poses, we wish to minimize:

$$\sum_{(s,t) \in P} \|\delta_s^t - d(\xi_s, \xi_t)\|_{\Lambda_s}^2$$

all poses ξ_i . P is the set of frame indices between which pose changes have been computed, and $\|\cdot\|_{\Lambda}$ is the Mahalanobis distance. Poses are parametrized using local rotations so that $d(\xi_s, \xi_t) = \xi_s - \xi_t$. Optimizing (2.1) involves solving a sparse linear system, which can be performed efficiently using conjugate gradient descent, for example. For more details, see [14].

3. 3D Model Reconstruction

Once frames have been globally registered, they are non-rigidly deformed during the reconstruction phase to produce a smooth triangular mesh.

To construct this mesh, frames are individually converted to meshes by using the pixel adjacency information in the original range scan. Each vertex q on a mesh is assigned the 3D location \mathbf{X} , surface normal \mathbf{n} and intensity I of its corresponding point in the registered scan. The uncertainty in these variables is computed by combining the effects of measurement uncertainty and registration error and stored along the other parameters:

$$q = \{ [\mathbf{X}_q \quad \mathbf{n}_q \quad I_q]; \Lambda_q \}$$

Reconstruction then involves a discretization of these vertices into a linked voxel space (described in the following section), followed by a merging of nearby voxels using the Cubic Ray Projection algorithm of section 3.2. The linked voxel space is finally converted to a triangular mesh and rendered.

3.1. Linked Voxel Space

To maintain an intermediate representation of the final 3D model, we use a voxel space. However, for our purposes, the simple voxel model has two main disadvantages: 1) the connectivity of meshes cannot be represented, and 2) converting this volumetric model to a mesh is difficult[8, 10]. To solve these problems, we use an augmented version of the voxel space called the linked voxel space.

In a linked voxel space, voxels maintain information about their connectivity beyond their immediate neighbors in the space. When converting a mesh to a linked voxel space, edges between vertices of the mesh are converted to links between voxels. In our representation, each voxel v is represented by a vertex q_v centered in the voxel and a list of links L_v , initially empty.

$$v = \{ q_v \quad L_v \}$$

After registration, each frame is converted to a mesh. The mesh is transformed to the pose recovered during the global registration phase, and accumulated into the linked voxel space.

The location of each vertex q in the mesh is quantized and mapped to a voxel v . This voxel v is updated as follows:

- The covariance Λ_v of v is updated with

$$\Lambda_v^{new} = \left([\Lambda_v^{old}]^{-1} + \Lambda_q^{-1} \right)^{-1}$$

- The mean surface normal \mathbf{n}_v at the voxel is updated with the normal \mathbf{n}_q of q using:

$$\mathbf{n}_v = \Lambda_v^{new} \left(\Lambda_q^{-1} \mathbf{n}_q + [\Lambda_v^{old}]^{-1} \mathbf{n}_v \right)$$

where Λ_q is the uncertainty in the node.

- The intensity value I_v is updated as follow

$$I_v = \Lambda_v^{new} \left(\Lambda_q^{-1} I_q + [\Lambda_v^{old}]^{-1} I_v \right)$$

- Each edge i of the vertex q points to a neighboring node q^i which is mapped to a voxel v^i . A link L_{v^i} is added to v if the voxel v^i is not already linked with v .

The mean surface normal of the voxels are used to guide the Cubic Ray Projection merging phase and ultimately become the normals of the final mesh model.

3.2. Cubic Ray Projection

The next stage of reconstruction thins out the voxel space by projecting voxels on one of the six faces of a cube that delimits the voxel space, merging voxels which fall on the same projection ray. As voxels are merged, the link information of the voxels is updated, resulting in a voxel space which can be trivially turned into a mesh.

This process aims to identify voxels which represent the same point on the object being modelled but which have been incorrectly registered by the registration process. We employ the heuristic that if two voxels are nearby, have the same normal, and lie along the same projection ray to the camera, they represent the same point and should be merged. The cube projection algorithm identifies such voxels by quantizing the normal vectors and providing an efficient data structure to aid the search. As a result, merging is an $O(n)$ algorithm, where n is the number of voxels.

Figure 1 depicts the merging process. The inverse of the covariance of a voxel is represented by the size of the dot. The arrow shows the direction of the normal. The highlighted line in the figure represent a projection ray to the left face of the cube. Along this ray, only voxels with a normal vector pointing in the direction of left face are selected. Voxels which are nearby and which are mapped to the same location on the surface of the cube are then merged. Merging two voxels involves updating one of them and unlinking the other.

The merging algorithm updates the mean normal, the intensity value and the adjacency information of the voxel with the lowest covariance v_1 . The voxel with the highest covariance v_2 is unlinked from the rest of the voxel space. The specifics of the update are similar to the discretization step: 1

- Average normal \mathbf{n}_1 , intensity I_1 and covariance Λ_1 are updated as described in section 3.1.
- Each links $L_{v^i}^2$ are added to v^1 if the voxel v^i is not already linked with v^1 .

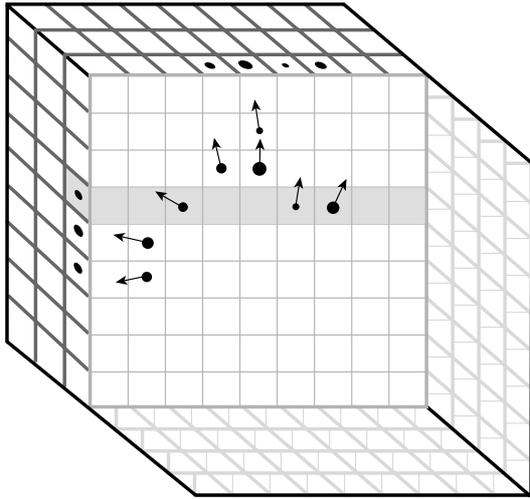


Figure 1. Result of voxels merging on one layer of the voxel cube. Occupied voxels are represented by a dot (inverse of the covariance) and an arrow (normal vector). The figure shows the projection of the voxels on two faces of the cube.

- All the links from v^i to v^j are removed during the unlinking stage. This step is possible since the voxel space is double linked.

Note that the update throws away only a small amount of information: it discards the voxel with the largest covariance, but updates the voxel with the lowest covariance with the former's normal and link information. This voxel merging step is in some ways similar to that of [5] where only one merging plane is used (instead of a cube) and all voxels along a ray contribute to a weighted average distance to the merging plane (instead of a weighted normal).

Merging all voxels which satisfy the merging criterion results in a representation where no two nearby voxels with similar normals project to the same face of the encompassing cube. This thinned spatial representation has associated adjacency information which make it readily available as a triangular mesh.

4. Results

When integrated with a real-time stereo camera, our system makes it possible to capture 3D models interactively and unobtrusively. Using SRI's Small Vision System, we captured about 10 seconds worth of range and intensity frames of a person moving in front of the camera. Figure 2 shows some typical frames from the sequence. The subject rotated his head from left to right, making a 70 degree arc about the vertical axis. Notice that the range information is missing for much of the face and must be recovered by incorporating many views.

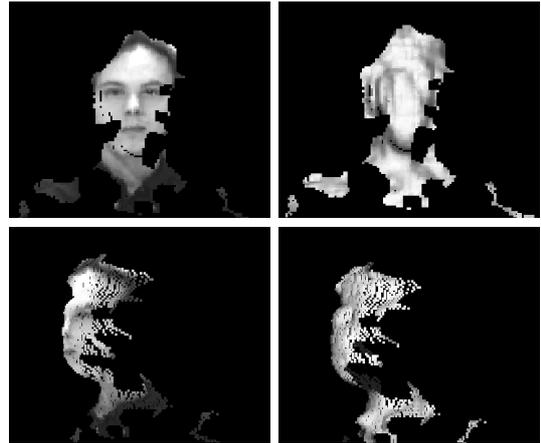


Figure 2. Sample images from the sequence. Left: Intensity images. Right: Depth images.

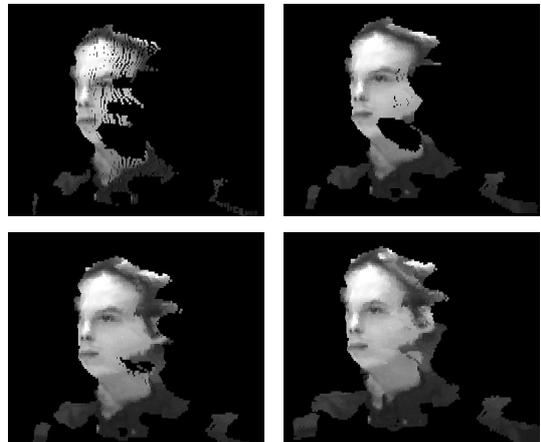


Figure 3. Progress of the model acquisition.

The ZBCCE registration step aligns all the 3D views together to create a dense 3D model. Figure 3 shows the progression of the 3D model during the registration step. We can observe that the face model is completed as the person turn his head. The registration runs online at the same time as the stereo processing, at about 7 fps on a 1.5Ghz Pentium 4.

The Cubic Ray Projection phase merges all the views into a linked voxel space. This step reduces the number of vertices from 200,000 to 18,000. Figure 4 shows the reconstructed 3D voxel space, along with the accompanying texture map. A solid arc of about 180 degrees was recovered from the 70 degrees of rotation. Global registration, 3D reconstruction, and rendering together took less than 1 second.

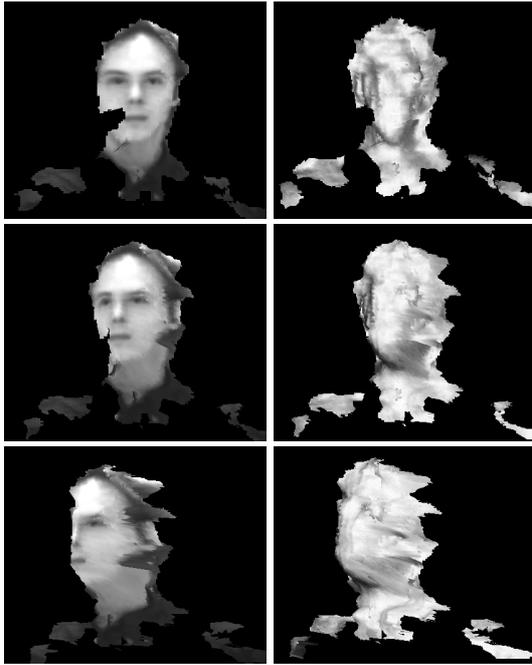


Figure 4. Final 3D mesh viewed from different directions.

5. Conclusion

We have demonstrated an efficient system for producing textured 3D models from range and intensity data. The system uses stereo cameras to obtain synchronized range and intensity frames, and hence does not require subsequent texture alignment. Our algorithm allows the object to be moved around freely to expose different views.

The frames first undergo a registration phase which computes relative pose estimates between pairs of frames, and globally solves for the optimal set of poses for all frames. Our registration algorithm uses range as well as intensity data in an image gradient-based approach, compensating for the poor quality of range from correlation-based stereo. The recovered poses are used to warp all frames to a canonical position, and a 3D model reconstruction step merges the registered frames together to build a 3D mesh of the object. We have demonstrated our system by reconstructing a model of a human head as the subject underwent a 70 degree rotation.

References

- [1] 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.
- [2] A.R. Bruss and B.K.P Horn. Passive navigation. In *Computer Graphics and Image Processing*, volume 21, pages 3–20, 1983.
- [3] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2724–2728, 1991.
- [4] B. Curless. From range scans to 3d models. *Computer Graphics*, 33(4), november 1999.
- [5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Computer Graphics*, 30(Annual Conference Series):303–312, 1996.
- [6] G.Turk and M. Levoy. Zippered polygon meshes form range images. In *Proceedings of SIGGRAPH vol. 2*, pages 311–318, 1994.
- [7] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill. 3d pose tracking with linear depth and brightness constraints. In *Proceedings of CVPR 99*, pages 206–213, Corfu, Greece, 1999.
- [8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Proceedings of SIGGRAPH*, 26(2):71–78, July 1992.
- [9] B.K.P. Horn and B.G. Schunck. Determining optical flow. *AI*, 17:185–203, 1981.
- [10] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Proceedings of SIGGRAPH*, 21(4):163–169, July 1987.
- [11] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4:333–349, 1997.
- [12] K. Pulli. Multiview registration for large data sets. In *Int.Conf. on 3D Digital Imaging and Modeling*, pages 160–168, 1999.
- [13] A. Rahimi, L-P. Morency, and T. Darrell. Reducing drift in parametric motion tracking. In *ICCV*, June 2001.
- [14] A. Rahimi, L.P. Morency, and T. Darrell. Reducing drift in parametric motion tracking. In *ICCV01*, volume 1, pages 315–322, 2001.
- [15] Harpreet S. Sawhney, Steve Hsu, and Rakesh Kumar. Robust video mosaicing through topology inference and local to global alignment. In *ECCV*, pages 103–119, 1998.
- [16] A. Stoddart and A. Hilton. Registration of multiple point sets. In *IJCV*, pages B40–44, 1996.