Dense 3D Modeling Using Consistency-Based Feature Point Evolution

Wende Zhang and Tsuhan Chen

Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
E-mail: {wendez, tsuhan}@andrew.cmu.edu

To render scenes by camera-based ambient intelligence, we propose a dense 3D modeling algorithm using feature points that dynamically appear and disappear over time; that is, we reconstruct a 3D model of the scene with a dense feature point set that "evolves" over time. As the scene's appearance changes due to camera movements, some existing feature points dynamically disappear while some new feature points dynamically appear relative to the camera. The newly generated feature points' 3D positions are initialized using nearby existing feature points' positions depending on their consistency including the distances in 2D image, stability and history. Our feature evolution, when incorporated into standard tracking and 3D reconstruction algorithms, provides more robust and denser 3D meshes. Consequently, the resulting 3D meshes and textures render novel 3D images better than meshes and textures produced using standard stereo techniques.

Keywords: Stereo vision; 3D modeling.

1. Introduction

We can apply camera-based ambient intelligence to model and render scenes, including those in art galleries. Static scene modeling includes stereo vision techniques and plenoptic function techniques. Stereo vision techniques exploit the difference in the images from different viewpoints.[1] By locating where points appear in both images (i.e., corresponding points), we can recover the underlying 3D locations of these points from epipolar geometry.[2] Instead of analyzing the 3D structure, plenoptic function techniques seek to estimate the light field in a region. The Lumigraph,[3] for example, bounds the concerned region with a cube and models the light passing through all points on that cube. With the static 3D scene reconstructed with any of the above approaches, synthetic images can then be rendered from different locations by interpolating between the multiple input images.

Dynamic scene modeling uses time series analysis, such as the Kalman filter,[4] Extended Kalman filter (EKF),[5] and particle filters.[6] Kalman filtering has been used to track feature points in video frames for reconstructing scenes.[7] EKF allows for nonlinear modeling for estimating the structure and motion of rigid objects, assuming smooth motion.[7] Particle filters provide nonlinear and non-Guassian noise modeling to feature point tracking.[6]

For modeling scenes, the dynamic tracking algorithms typically use a static set of feature points/patches, which may not remain reliable as the scene evolves. Ref. 8 provides a mechanism for generating and deleting feature points as they appear or leave the scene. However, it focuses on tracking a sparse feature point set only of dominant features for scene modeling. Ref. 8 initializes the state of each new feature point typically using the state of its single nearest neighbor. In contrast, for high-quality rendering, our work not only tracks a dense 3D point set of both dominant and subtle features, but also initializes their underlying states (especially for subtle features) using the existing prior knowledge, the tracking results of their multiple neighboring states based on their consistency including distances in image, stability and history. The final result is dense modeling of the 3D geometry of a scene from one moving camera.

The paper is organized as follows. In the next section,we discuss our dense 3D modeling using consistency-based feature point evolution. The experimental results are then discussed in Section 3; our proposed method outperforms the standard stereo technique in rendering both synthetic and real images. Finally, concluding remarks are given in Section 4.

2. Modeling Scenes by Consistency-Based Evolution

Evolution is a dynamic feature point extractor embedded in standard time-series analysis (see Figure 1). As video progresses over time, certain tracked 3D points (e.g., state X_t) will have noisy feature points Y_t that become difficult to track while, conversely, new 3D points will appear which are robust and easy-to-track. Hence, we only model each portion of the scene while it is easy to track. In addition to proposing which feature points (and associated states) to model at each time frame, we also propose a novel consistency-based "state passing" mechanism that initializes the states of the newly generated feature points in each frame. Evolution proceeds as follows, where Steps 1, 2, & 3 below correspond to standard EKF, and where Steps 4 & 5 below are the key contributions of this work:

(1) Initialization (time $t = 0$ only): Find feature point set $\{Y_0\}$ at time $t = 0$, using a Harris corner detector.[9] Let $\{X_0\}$ be the set of corresponding

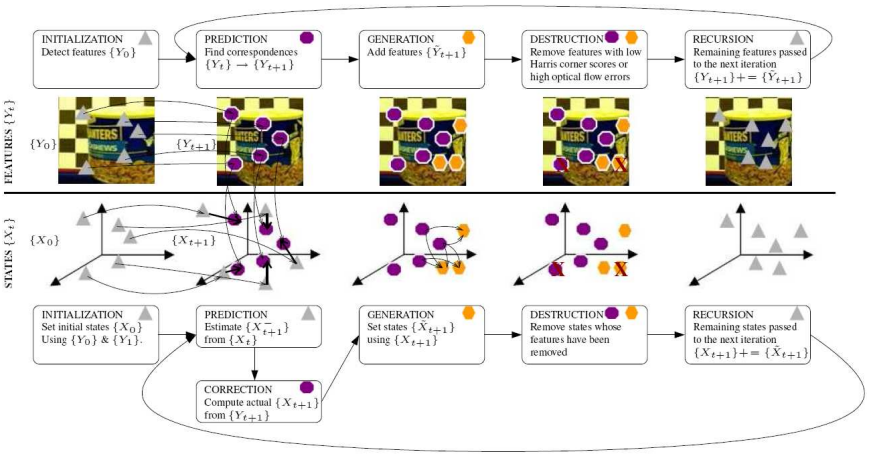

Fig. 1. Evolution Flow Chart. The evolution of the features is modeled on top: images with sample feature points are marked. The respective evolution of the states is modeled on bottom: the estimated 3D positions of the sample feature points are plotted. INITIALIZATION, PREDICTION, and CORRECTION follow standard EKF while GENERATION and DESTRUCTION follow our proposed evolution framework.

states. For each $y_0 \in Y_0$, initialize its filter's state $x_0 \in \{X_0\}$:

$$x_0 = \left[y_0[1], y_0[2], \text{depth}(y_0, y_1)\right]^T \tag{1}$$

where the three elements represent the horizontal, vertical, and depth positions, respectively, of y_0. $\text{depth}(y_0, y_1)$ is the depth obtained using stereo vision on $\{Y_0\}$ and $\{Y_1\}$. $\{Y_1\}$ are the corresponding positions of $\{Y_0\}$ in frame 1 using pyramid optical flow.[10]

(2) Prediction: For each $x_t \in X_t$, predict x_{t+1} by EKF's state equation.

$$X_t = R_t X_{t-1} + T_t + q_t \tag{2}$$

where R_t and T_t are the rotation matrix and the translation vector between the two camera viewpoints. q_t is the noise of state transition. For each $y_t \in Y_t$, find its corresponding position y_{t+1} in frame $t+1$ using pyramid optical flow.[10]

(3) Correction: For each $x_{t+1} \in X_{t+1}$, correct its value by EKF's observation equation.

$$Y_t = f_t(X_t) + r_t = \begin{bmatrix} X_{t,1}/X_{t,3} \\ X_{t,2}/X_{t,3} \end{bmatrix} + r_t, \tag{3}$$

where r_t is the noise of observation.

(4) Consistency-Based Generation: Find feature point set $\{\tilde{Y}_{t+1}\}$ using a Harris corner detector. These feature points are chosen independently of

the predicted $\{Y_{t+1}\}$. For each of the new feature points $\tilde{y}_{t+1} \in \{\tilde{Y}_{t+1}\}$, find its corresponding position $\tilde{y}_t$ in frame t using (reverse) pyramid optical flow. Let $\{\tilde{X}_{t+1}\}$ be their (uninitialized) states.
Initialize each new state $\tilde{x}_{t+1} \in \{\tilde{X}_{t+1}\}$: Let $\tilde{x}_{t+1}$ be initialized using consistency-based weights $w_{x'_{t+1}}$ on the nearby existing states $X'_{t+1} = \{x_{t+1} \in X_{t+1} \,|\, \|y_{t+1} - \tilde{y}_{t+1}\| < Th_y\}$:

$$w_{x'_{t+1}} = \frac{\text{Age}(x'_{t+1})}{\|y'_{t+1} - \tilde{y}_{t+1}\| \sum_{i=t+1-\text{Age}(x'_{t+1})}^{t+1} \alpha^{t+1-i} |E_i(x'_i)|^2}$$

$$\tilde{x}_{t+1} = \frac{\sum_{x'_{t+1} \in X'_{t+1}} w_{x'_{t+1}} x'_{t+1} + \beta X_0}{\sum_{x'_{t+1} \in X'_{t+1}} w_{x'_{t+1}} + \beta}, \tag{4}$$

where $\tilde{y}_t$, y_t, and y'_t are the observations of $\tilde{x}_t$, x_t, and x'_t, respectively; Th_y is the distance threshold for defining a new state's neighbors in the 2D image; β is the weight on the prior state position X_0; α_t is the weight on stability, $E_t(x'_t)$ is the EKF correction error at time t; and $\text{Age}(x'_{t+1})$ represents feature point's history, which is the number of frames that state x'_{t+1} has been in existence.

(5) Destruction: Define $\mathcal{P}_t(y'_t)$ as the square patch centered at pixel y'_t in frame t. Determine optical flow matching errors for existing $y_{t+1} \in \{Y_{t+1}\}$ and for new feature points $\tilde{y}_{t+1} \in \{\tilde{Y}_{t+1}\}$, respectively:

$$E_{t+1}(y_t, y_{t+1}) = \|\mathcal{P}_t(y_t) - \mathcal{P}_{t+1}(y_{t+1})\| \tag{5}$$

$$E_{t+1}(\tilde{y}_{t+1}, \tilde{y}_t) = \|\mathcal{P}_{t+1}(\tilde{y}_{t+1}) - \mathcal{P}_t(\tilde{y}_t)\| \tag{6}$$

Define $\text{HC}_t(y_t)$ as the corner score returned by the Harris corner detector for feature point y_t at time t. Destroy any existing feature point y_{t+1} or new feature point $\tilde{y}_{t+1}$ that fails either of its respective tests:

$$y_{t+1} : E_{t+1}(y_t, y_{t+1}) < Th_E \tag{7}$$

$$\text{HC}_{t+1}(y_{t+1}) > Th_{HC} \tag{8}$$

$$\tilde{y}_{t+1} : E_{t+1}(\tilde{y}_{t+1}, \tilde{y}_t) < Th_E \tag{9}$$

$$\text{HC}_{t+1}(\tilde{y}_{t+1}) > Th_{HC}, \tag{10}$$

where Th_E is the threshold for optimal flow matching error and Th_{HC} is the threshold for corner score. Let the sets of states of feature points for the next iteration be:

$$\{X_{t+1}\} = \{X_{t+1}\} + \{\tilde{X}_{t+1}\} \tag{11}$$

$$\{Y_{t+1}\} = \{Y_{t+1}\} + \{\tilde{Y}_{t+1}\}. \tag{12}$$

In summary, we detect additional new feature points in every frame. Instead of initializing the state of the newly generated feature points from scratch, we borrow information from their neighbors based on consistency. Only the feature points with good 2D correspondences (large corner scores and low matching errors) can be passed on to the next iteration. We also let feature points die if they do not have good 2D correspondence across neighboring frames since we cannot accurately reconstruct their 3D locations anyways.

Once the tracking of the evolving dense feature points is complete, the underlying states can then be used to construct the 3D mesh. First, as we are tracking the feature points using the consistency-based evolution, we utilize the EKF's correction errors to remove those feature points which were poorly tracked. Then, the remaining, dense reliable states are used to build the 3D mesh using Delaunay triangulation[11] on the states.

3. Experiments

We compare the results of our proposed scene modeling method to standard stereo modeling. With stereo, we first identify the feature points using a Harris corner detector in the current image and then apply pyramid optical flow [10] to find their corresponding positions in the next image. Given the intrinsic and extrinsic camera parameters, we then find the feature point's 3D locations using a least squares estimation for future rendering.

We first ran experiments on synthetic data. A toy car is captured by a moving camera simulated by POV-ray.[12] As illustrated in Figure 2, 360 images are captured at the resolution of 320×240 pixels with known intrinsic and extrinsic camera calibration parameters as the input image sequence. As shown in Figure 3, we first reconstructed the feature points of the scene using both our proposed algorithm and the standard stereo technique. Based on the 3D feature points, we then built the triangle mesh of the scene at each viewpoint. We rendered the scene at different viewpoints using the view-dependent meshes and textures corresponding to the closest viewpoint from the sample images. In the synthetic results, the proposed method has a better rendering quality; for example, the rear surface of the car is smoother as shown in Figure 3. The proposed results had denser feature points than the stereo technique because it kept the well-estimated feature points from the previous frames. To have a fair comparison, we could also increase the number of feature points by lowering the threshold of the corner detector. However, those feature points with small corner values would have less reliable 2D correspondences and less reliable 3D reconstruction. Therefore, they would result in a noisier 3D geometry.

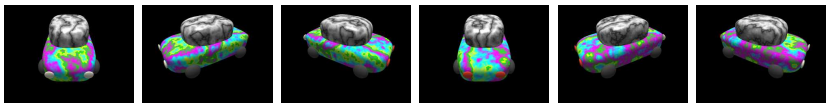

Fig. 2. Sample input images from the synthetic scene

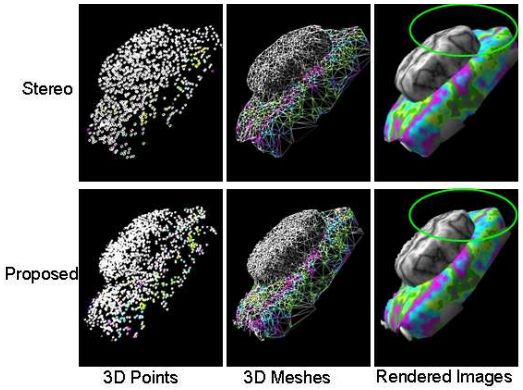

Fig. 3. The reconstruction of the synthetic scene. The rendering of the proposed method is better compared to stereo, as illustrated by the ovals.

We also performed experiments on real data. A peanut can with a checker board is captured by a moving Canon G5 camera with pre-calibrated intrinsic camera parameters. The peanut can, tissue box and checkerboard are overlapped in the scene. Therefore, the scene depth map is not smooth, especially at the object boundaries. As illustrated in Figure 4, 35 images are captured at the resolution of 2592×1944 pixels. The extrinsic camera calibration parameters can be derived from the checker board pattern. Using results both from stereo and from the proposed method, we also rendered images at different viewpoints, as shown in Figure 5. The rendering with the proposed method is better compared to the stereo technique because of the reliable and dense 3D reconstruction; notice the difference in the rendering of the peanut can lid.

4. Conclusion

We proposed a dense feature point reconstruction algorithm that exploits the characteristics of dynamic, evolving scenes. When the video gets new perspectives on the scene, new and reliable feature points will appear and be tracked. As these new feature points may be in the vicinity of existing feature points, the new feature points' states can be estimated using the refined estimates of the existing feature points' states based on their consistency including distances in image, reliability and history. The result is a

Fig. 4. Sample input images from the real scene

Fig. 5. The rendering results of the real scene. The rendering of the proposed method is better compared to stereo, as illustrated by the ovals.

dense 3D model that is better constructed than using existing reconstruction techniques.

References

1. D. Scharstein and R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *IJCV* **47,**2002.
2. Y. Lu, J. Z. Zhang, Q. M. J. Wu and Z. N. Li, A survey of motion-parallax-based 3-d reconstruction algorithms, *IEEE Transanctions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **34,**2004.
3. S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, The Lumigraph, in *Computer Graphics (SIGGRAPH '96)*, 1996.
4. R. E. Kalman, A new approach to linear filtering and prediction problems, *Transactions of the ASME–Journal of Basic Engineering* **82,**March 1960.
5. H. Cox, On the estimation of state variables and parameters for noisy dynamic systems, *IEEE Transactions on Automatic Control* **9,**January 1964.
6. S. K. Zhou, R. Chellappa and B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, *IEEE Transactions on Image Processing* **13,**November 2004.
7. T. J. Broida, S. Chandrashekhar and R. Chellappa, Recursive 3-D motion estimation from a monocular image sequence, *IEEE Transactions on Aerospace and Electronic Systems* **26,**July 1990.
8. M. Trajkovic and M. Hedley, Robust recursive structure and motion recovery under affine projection, in *British Machine Vision Conference*, 1997.
9. C. Harris and M. Stephens, A combined corner and edge detector, in *4th Alvey Vision Conference*, 1988.
10. B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in *IJCAI '81*, (Vancouver, 1981).
11. A. Okabe, B. Boots and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* (Wiley, New York, 1992).
12. Persistence of vision ray tracer (POV-Ray) http://www.povray.org/.