Editing Real World Scenes: Augmented Reality with Image-based Rendering

Dana Cobzas, Martin Jägersand, Keith Yerex
CS, University of Alberta, Edmonton, AB, Canada, T6G2E8
www.cs.ualberta.ca/~ {dana,jag,keith}

Abstract

We present a method that using only an uncalibrated camera allows the capture of object geometry and appearance, and then at a later stage registration and AR overlay into a new scene. Using only image information first a coarse object geometry is obtained using structure-from-motion, then a dynamic, view dependent texture is estimated to account for the differences between the reprojected coarse model and the training images. In AR rendering, the object structure is interactively aligned in one frame by the user, object and scene structure is registered, and rendered in subsequent frames by a virtual scene camera, with parameters estimated from real-time visual tracking. Using the same viewing geometry for both object acquisition, registration, and rendering ensures consistency and minimizes errors.

1 Introduction

In Augmented Reality (AR) virtual objects are registered with and overlaid into a real scene[1]. Recently, visual tracking of image features was used to establish camera pose and register the virtual object with the camera view [4, 5]. Most current systems allow only augmentation with synthetic virtual objects which have been manually defined and represented as Euclidean computer graphics (CG) models. We present a method which allow image based capture of appearance and geometry of real objects and then, at a later stage, the insertion and registration of these objects with a new scene. The same visual tracking and structure-from-motion (SFM) methods are used for both the objects and scene to yield a consistent representation and minimize re-projection errors when the AR object is reprojected through the estimated scene camera matrix. Unlike other non-Euclidean approaches, in the registration stage, we upgrade the estimated object structure to scaled Euclidean by alignment with the camera plane. This allows a natural interface where the user can specify the desired object alignment using metric angles instead of having to construct affine[4] or projective[5] basis alignments in pairs of images. To capture the appearance of the object we developed an image-based error correction method using *dynamic textures* that compensates for the sparse geometry.

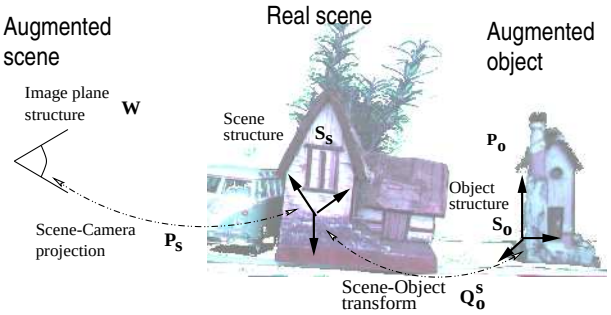

Figure 1: Coordinate frames in the AR system

2 Theory

We represent objects and scenes as in computer graphics using texture mapping on a triangular mesh with n (homogeneous coordinate) node points $S = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n]$. Geometrically, the AR registration can be described as the alignment of an introduced object structure S_o with the scene structure S_s, so that the augmented structure is $S_a = [S_s; Q_o^s S_o]$, the concatenation of the scene structure with the object structure transformed into the scene coordinates. See Fig. 1. In a fully calibrated system Q is a Euclidean transform defined by the user. In a vision based systems Q can be also an affine transform [4] or projective [5].

2.1 AR Object Capture

Geometry Using an SFM algorithm [6] we recover object 3D structure S_o from m training images and tracked image points $(I^1, p^1), \ldots, (I^m, p^m)$, where $p_j^i,\ i = 1 \ldots m, j = 1 \ldots n$ contains the coordinates of tracked point j in view i. Now this structure in an arbitrary image k, given an affine camera P^k represented as a 2×4 is reprojected into a new view $p^k = P^k S_o$

Dynamic texture Generating high quality renderings using standard texture mapping of objects captured by SFM has proven difficult. In practice, relatively few points can be reliably tracked, and the resulting object model is sparse and triangles are not perfectly aligned with the true object. The resulting texture variation for view k can be written to first order as $\mathbf{T}^k = \mathbf{T}^0 + B\mathbf{y}^k$, where $\mathbf{T}^0$ is a column flattened reference (mean) texture, and B is the texture derivatives w.r.t. pose changes $\mathbf{y}$. Elsewhere we have shown how to estimate B from image variability in the training

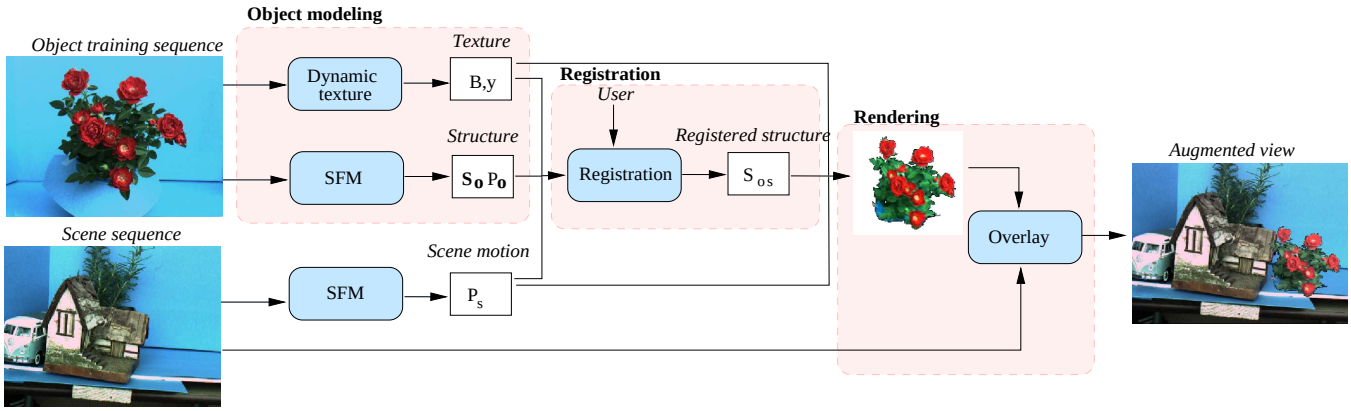

Figure 2: Overview of the AR system. An object is captured using SFM and represented as a sparse affine 3D model S and dynamic texture basis B. In AR rendering it is aligned with and reprojected into a tracked scene.

set[2]. In the rendering stage instead of using only one texture image, a continuously time varying texture is modulated for each view, intuitively playing a small "movie" on each model triangle, where the "movie" accounts for the difference in detail and alignment errors with the true object.

2.2 Scene capture and object alignment

The scene geometry is captured and tracked using the same SFM. To allow a metric object alignment the alignment key frame is upgraded to a scaled Euclidean model by orthogonalizing the affine camera w.r.t. the image coordinate system[2] giving a projection matrix of the form:

$$P^k = [s^k R^k | \mathbf{t}^k] \tag{1}$$

Only one frame is needed to align the object by interactively controlling the rotation R, translation t and scale s. The now registered object structure is:

$$S^w_{os} = \frac{1}{s^r_s} R^{rT}_s (s_o R_o S_o + \mathbf{t}_o - \mathbf{t}^r_s) \tag{2}$$

where $s^r_s, R^r_s, \mathbf{t}^r_s$ represents the motion of the reference frame, S_o the structure of the object, and $s_o, R_o, \mathbf{t}_o$ the desired motion of the object.

Figure 3: Augmented scene with real and added house.

2.3 Rendering

For rendering the augmented scene we track the scene points in real-time and compute the scene projection matrix P^k_s. The registered object structure S_{os} is then projected using $p^k_o = P^k_s S_{os}$ and textured with the modulated dynamic texture T^k.

We have implemented our system on standard consumer-grade PC's in c++, Open-GL with NVidia specific extensions to handle the dynamic texture blending in HW, and Matlab for the user interface. For real-time image tracking we have extended XVision[3] and interfaced it to our geometry code. The system is capable of rendering 3-4 AR objects at 30Hz on a 1.9GHz P4 laptop with GeForce4-to-go graphics.

3 Results and Discussion

Figure 3 presents results of the augmented scene from Figure 2 augmented with a separately captured house object. Being able to capture AR objects from camera images extends the applicability of AR to combining real objects in new views with real scenes. Since the same viewing geometry is used for both scene and object capture there can be less alignment error than when combining an affine scene registration with a Euclidean object. We found that the traditional problems of sparse inaccurate structure from SFM was effectively compensated for using the dynamic view dependent texture modulation.

References

[1] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355.

[2] D. Cobzas and M. Jagersand. Tracking and rendering using dynamic textures on geometric structure from motion. In *ECCV*, 2002.

[3] G. D. Hager and K. Toyama. X vision: A portable substrate for real-time vision applications.

[4] K. N. Kutulakos and J. R. Vallino. Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):1.

[5] Y. Seo and K. Hong. Calibration-free augmented reality in perspective. *IEEE Transactions on Visualization and Computer Graphics*, 6(4), 200.

[6] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.