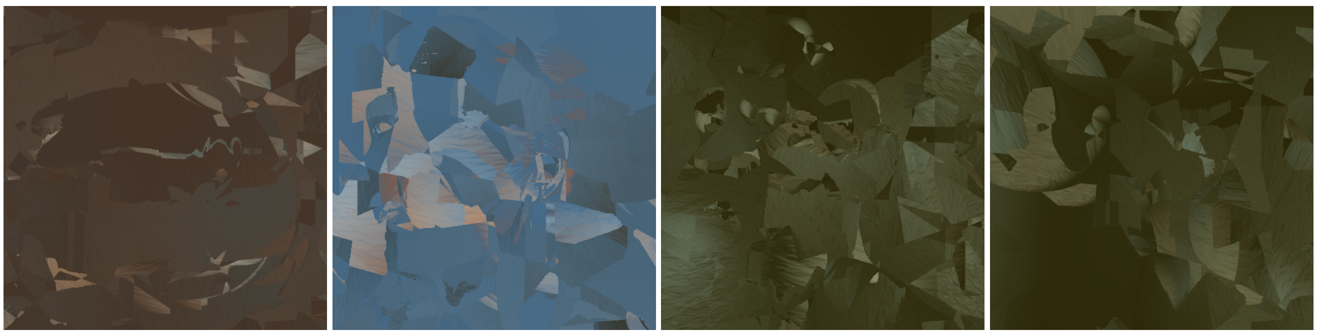
Remote Empathetic Viewpoint: A Novel Approach To Extending Cubism

Chris Morrison
Department of Visualization, Texas A&M University
College Station, Texas
cmorr@tamu.edu

Ergun Akleman
Departments of Visualization & Computer Science and Engineering, Texas A&M University
College Station, Texas
ergun.akleman@gmail.com



(a) Colors sampled from George Braque - Still Life with Banderillas

(b) Colors sampled (priority reversed) from Juan Gris - Portrait of Picasso

(c) Colors sampled from George Braque - Violin and Pitcher

(d) Colors sampled from George Braque - Violin and Pitcher

Figure 1: *Example frames taken from animated cubist renderings using the Remote Empathetic Viewpoints approach.*

ABSTRACT

In this paper, we present the concept of Remote Empathetic Viewpoints, an approach which allows the viewer to simultaneously access multiple remote viewpoints, granting us the ability to explore and extend the concepts of Cubism in 3D and in animation. In Remote Empathetic Viewpoints, we utilize a single Primary View Camera, a small set of Control lights, and a significantly large number of secondary cameras, whose positions and directions are referenced by Control Lights. By using such a multi camera system, we overcome the “shower door” affect that comes from using multiple cameras, which are used to obtain cubist rendering. By animating camera movement, we obtain temporal cubist art.

CCS CONCEPTS

• **Computing methodologies → Non-Photorealistic Rendering**.

KEYWORDS

Rendering, Non-Photorealistic Rendering, Cubism, Multi-Camera Rendering, Remote Empathetic Viewpoint, Generative

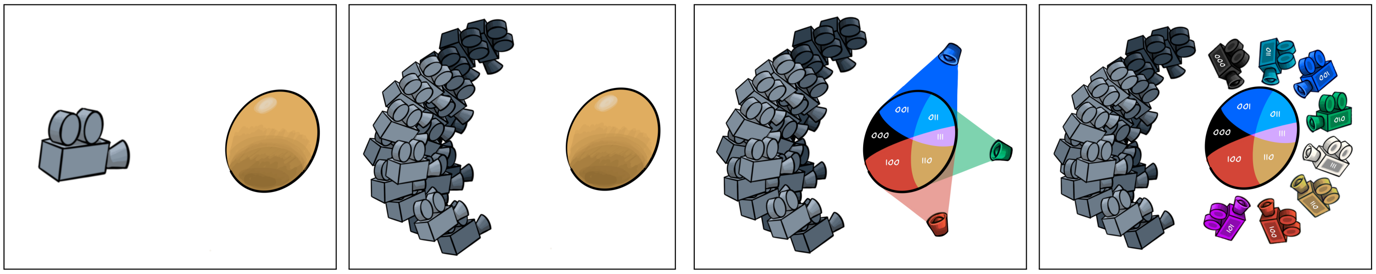
SIGGRAPH '20 Posters, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7973-1/20/08.
https://doi.org/10.1145/3388770.3407404

ACM Reference Format:
Chris Morrison and Ergun Akleman . 2020. Remote Empathetic Viewpoint: A Novel Approach To Extending Cubism. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '20 Posters), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388770.3407404

1 INTRODUCTION AND MOTIVATION

We, as humans, are confined to a single visual perspective (see Figure 2a). Many conventions of picture making in representational art is originally developed based on this limitation in perspective. However, in the nineteenth century, the conventions including perspective started to break down in art. A significant departure from traditional perspective came from the Cubist painters, who depicted objects from many sides at once using multiple views (see Figure 2b). In computer graphics, Glassner devised a free-form "Cubist" camera system using a commercial 3D modeling and rendering package [Glassner 2000]. His system uses ray tracing to render images, taking one unique viewpoint for each pixel in the final image. Meadows and Akleman developed another ray-tracing based cubist rendering approach, known as camera painting [Meadows and Akleman 2000]. They used the color information from digital images to distort 3D scenes rendered with ray tracing. Jeff Smith et al. developed an interactive cubist rendering system based on OpenGL, called MultiCam system [Smith et al. 2004]. This system uses Glassner's method of two NURBS planes, one designated as the "eye" plane and another as the "lens" plane. The main problem with these methods, they are only suitable to create a single cubist image. If we create dynamic paintings that consist of successive renderings,



(a) Classical Ray Tracing Set-Up. **(b) Multi-Camera Set-Up.** **(c) Control-Light Set-Up.** **(d) Our complete set-up.**

Figure 2: This image sequence demonstrates our process in a nut-shell. The classical ray tracing set-up consists of a single "primary" camera and a single scene as shown Figure 2a. In Cubist camera setting, the original camera is replaced by multiple "primary" cameras (see Figure 2b). Even if we change primary camera parameters, this set up creates a shower door effect. To avoid this effect, we also use multi-scene. To create a multi-scene, we first decompose the scene into up to 2^k different regions using control lights as shown in Figure 2c. Then each regions is rendered using an associated "secondary" camera as shown in Figure 2d.

we end-up with shower door effect. In other words, the viewers feel as if they are looking at the scene through a shower door. This is hard to resolve even if we change the camera positions during the animation. In this work, we provided a solution by decomposing the scene using control lights. One additional contribution of this paper is to use Barycentric shading to quickly obtain the look-and-feel of any given cubist painter.

2 PROCESS

In this work, we have developed a process to obtain animated cubist renderings. Our process utilizes a primary and secondary camera setup that broadly connects the viewpoints of the two setups via some pre-defined rules. Our process of obtaining dynamic cubists paintings consists of four stages: (1) Primary Camera Set-Up Using Camera Painting; (2) Scene Decomposition using Control Lights; (3) Secondary Camera Set-Up Based on Scene Decomposition; and (4) Emulating Cubist Color Palettes using Barycentric Shading. Each of these stages is simple and intuitive. Any 2D artist with minor training in 3D modeling and animation can potentially produce a dynamic cubist paintings using this process.

Primary Camera Set-Up Using Camera Painting: To set up primary cubist camera, we use camera painting. Let $\boldsymbol{p}_e = (x_e, y_e, z_e)$ denote the 3D position of the eye and $\boldsymbol{p}_p(u, v)$ denote the 3D position of a pixel (u, v), which corresponds a 2D position in the final image. In camera painting, we have a control image, which is the same size of the original image. Let $\boldsymbol{c}(u, v) = (r(u, v), g(u, v), b(u, v))$ denote color of the pixel (u, v), where r, g and b are numbers between 0 and 1. Let a vector $\vec{v}$ computed as $\vec{v} = (2r(u, v) - 1, 2g(u, v) - 1, 2b(u, v) - 1)$. Using this vector, we recompute the position of the pixel as $\boldsymbol{p}'_p(u, v) = \boldsymbol{p}_p(u, v) + s\vec{v}$ where s the scale parameter. In initial render pass, we ray-cast the scene by sending rays from $\boldsymbol{p}_e$ in the direction of $\boldsymbol{p}'_p(u, v)$. Note that The value of $s = 0$ corresponds to single primary camera. The larger s is, the more "multi-perspective" effect is pronounced.

Scene Decomposition using Control Lights: We decompose the scene in initial render pass by using packed colors. Let $C = 2^{16}R + 2^8 B + G$ where R, G, and B between 0 and $28 - 1 = 255$ denote a packed color. Also, let $C_i = 2^i$ denote the packed color of the control light i where $i = 0, 1, \ldots, K - 1$. Now, let $a_i(u, v)$ denote the visibility of light i from the shading point that is visible from pixel (u, v), then we compute $a_i(u, v)$ as follow

$$a_i = \begin{cases} 1 & \text{if light } i \text{ is visible in shading point} \\ 0 & \text{otherwise} \end{cases}$$

Then the packed color of pixel (u, v) in the first rendering pass is computed as $C(u, v) = \sum_{i=0}^{K-1} a_i L_i$ as shown in Figure 2d: This process gives us a integer between 0 and $2K - 1$.

Secondary Camera Set-Up Based on Scene Decomposition: We create 2^K randomly created cameras. Let $\boldsymbol{p}_{j,e}$ denote the 3D position of the eye and $\boldsymbol{p}_{j,p}(u, v)$ denote the 3D position of a pixel (u, v), which corresponds a 2D position in the final image for secondary camera j where $j = 0, 1, \ldots, 2^{K-1}$. Then, to compute the color of pixel (u, v), during second render pass we simply use the camera $j = \boldsymbol{C}(u, v)$, where $\boldsymbol{C}(u, v)$ is the packed color computed in the the first pass.

Emulating Cubist Color Palettes using Barycentric Shading: To obtain desired look-and-feel, we use Barycentric shader. We simply obtain dark, $\boldsymbol{c}_0$, and light, $\boldsymbol{c}_1$, colors from actual cubist paintings and based on the diffuse illumination variable $t = 0.5(\cos\theta + 1)$, we compute final color as $\boldsymbol{c} = \boldsymbol{c}_0(1 - t) + \boldsymbol{c}_1 t$ The shading stage guarantees to obtain desired look-and-feel. Moreover, since this process effectively decouples first and second rendering passes, we do not see shower door effect anymore.

REFERENCES

Andrew S Glassner. 2000. Cubism and cameras: Free-form optics for computer graphics. *Microsoft Research MSR-TR-2000* 5 (2000), 2.

Scott Meadows and Ergun Akleman. 2000. Creating Abstract Digital Paintings with Painting Camera Technique. In *2000 IEEE Conference on Multimedia. An International Conference on Computer Visualization and Graphics*. IEEE, NewYork, NY, 250–254.

Jeffrey Smith, Ergun Akleman, Richard Davison, John Keyser, et al. 2004. Multicam: A system for interactive rendering of abstract digital images. In *Bridges: Mathematical Connections in Art, Music, and Science*. Bridges Conference, Arizona, 265–272.