

Augmented reality - Applying noise and blur to the virtual content to make it appear as real objects

- Anuj Bhatnagar and Sreesudhan Ramakrish Ramkumar

Motivation:

Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual / virtual objects. One way of experiencing augmented reality is using a handheld device like a smartphone or a tablet and streaming camera feed to the display while augmenting virtual object in it. This is also called as video-passthrough experience. The user views the content rendered to the display where camera feed is rendered from the camera sensor mounted to the device along with the virtual object rendered on top of the camera preview. The cameras present in the handheld devices are small since they need to fit within the form factor of the handheld device. So, the image quality of the camera preview is compromised. The noise in camera preview also worsens if the environment is poorly lit. When a virtual object is rendered in the camera preview, the virtual object looks perfect since the model is generated by a computer. This kills the experience since the virtual object doesn't blend in the surrounding environment and looks obvious to the human eyes that it is artificial. One way of enhancing the experience is to identify the noise present in the camera preview and applying similar noise level to the virtual object so that the virtual object looks like a real object. Another issue with the camera in a smartphone is the limited depth of field within which objects look sharp and focused when viewed through the camera preview on the display of the handheld device. In general, the depth of field of a smartphone camera is between ~20 cm to ~2 meters. When the camera is focussed to an object in the near plane, the objects in the infinity are blurred in the camera preview. When a virtual object is rendered near the far plane, they don't blend with the surrounding environment since the real objects surrounding it look blurred. To improve the visual perception, the virtual object can also be blurred depending on the focus distance being rendered.

Related work:

Camera noise estimation:

Ryan et al (1) have analyzed different noise patterns of CMOS sensor in detail including its source and effects on image quality. They have mathematically modeled them.

Yeul-Min Baek et al (2) have analyzed different noise patterns and have modeled the integrated noise coming from all the sources.

C. Manders et al (3) have modeled the sensor noise distribution based on images captured with different illumination of same scene.

Project overview / goals:

As part of our project, We plan to render the noise and blur to the virtual objects using the noise and blur techniques that were taught in the course lecture (instead of using the APIs). This involves following steps.

Noise:

- To render noise to the virtual object, We need to understand the noise profile of the camera feed. We plan to analyze the camera image and estimate the noise profile. This would require sampling the texture in the camera feed and looking for consistent and uniform regions, determining the difference in pixel strength and color across the uniform region. We also plan to use the ISO and lux value provided by the metadata of the camera feed to understand the

amount of brightness strength added by the image processing pipeline. This provides some clue about the lighting condition of the environment.

- When the object is rendered, the noise profile extracted from the camera feed is applied.

Blur:

- To render blur to the virtual object, We need to understand the current focal distance at which the camera is focussed. Along with this, we also need to understand the depth of field of the camera. These parameters can be obtained from the metadata of the camera feed. When a virtual object is rendered, the focus distance (z-depth assuming camera is the origin) at which the object is rendered should also be taken into account.
- When the object is rendered, a blur filter is applied where the blur strength depends on the focus distance of the object, focus distance at which the camera is focussed and the depth of field of the camera.

Programming language / development environment / framework:

OS - iOS

Hardware - iPhone or iPad

Development environment - Xcode

Programming language - Objective C, Swift, Metal

Frameworks - AVFoundation (to retrieve camera feed from the device)

Renderer - Custom. We plan to write a simple renderer that overlays a virtual object against the camera backdrop.

Milestones:

- Understand and survey different noise estimation techniques. Pick a technique that would model the integrated noise.
- Implement / Model noise estimation from a real camera feed.
- Apply noise model to the rendered object.
- Understand focus distance and capture focus distance and depth of field of the camera feed while in auto focus mode.
- Apply blur technique based on focus distance of the scene, depth of field and focus distance at the virtual object is rendered.

Timeline & Goals:

- Implement noise estimation (02/19)
- Implement noise injection to rendered scene (02/26)
- Implement blur estimation (02/1)
- Implement blur introduction (02/3)
- Prepare final poster and documentation (02/10)

References:

- (1) <https://ieeexplore.ieee.org/abstract/document/4215175> (A Comprehensive Tool for Modeling CMOS Image-Sensor-Noise Performance)
- (2) https://link.springer.com/chapter/10.1007/978-3-642-01811-4_37 (Integrated Noise Modeling for Image Sensor Using Bayer Domain Images)
- (3) <https://ieeexplore.ieee.org/document/1689264> (Digital Camera Sensor Noise Estimation from Different Illuminations of Identical Subject Matter)
- (4) <https://developer.apple.com/documentation/arkit/arframe/3255173-cameragraintexture?language=objc>
- (5) <https://developer.apple.com/documentation/scenekit/scncamera/2881741-wantsdepthoffield?language=objc>
- (6) <https://developer.apple.com/videos/play/wwdc2017/604/>