Lurch!: Interactive Rendering Pipeline Automation

Alexander Kolliopoulos
Pixar Animation Studios

The final rendering process on a feature film requires tracking and coordinating work on several hundred shots in a small span of time. This typically requires running several quality assurance tests, checking that shots meet specific performance gates, submitting and monitoring renders, and comparing renders to bless images from upstream departments. We present a system for automating this in a way that gracefully handles failure cases by socially engaging rendering TDs.

1 The Problem

A rendering department on a feature film is responsible for producing the final deliverable frames that will appear in the film. The process of generating these frames is complex and has many points of failure, but deadlines are tight, and inventory must be processed quickly. A feature rendering department is typically staffed by a small number of technical directors (TDs) who are responsible for processing a high volume of shots as they are delivered by upstream departments. Each TD might be responsible for tens of shots in a week, making time a premium that is best spent on the shots that need work the most. We present an automation solution used on *Monsters University* to streamline the rendering department workflow.

Figure 1: *A feature rendering department pipeline that every shot goes through. Rendering TDs are responsible for running and monitoring many steps in the process.*

2 The Traditional Approach

On past films, the rendering pipeline at Pixar has been run fully manually. A coordinator, manager, and TDs would monitor the database and renderfarm to see when work is available; new work would be assigned by a coordinator or manager; and QA/rendering commands were run manually with output inspected by a human at every step.

In some cases, such as converting already-completed films to 3D, renders have been submitted by an automated process, and the final renders were automatically compared to known-good renders (from the completed mono film). These bots interacted with a team passively, through setting database statuses.

3 Our Solution: An Interactive Render Bot

We structured our rendering pipeline around a bot that is capable of fully processing *clean* shots–those requiring no intervention from a render TD–while intereacting with TDs through a chat channel to reassign shots that it is unable to complete. A set of scripts handles the automated processes of the render pipeline, while an IRC chat bot component interacts with the team in an IRC channel to give important status updates and redistribute shot work that cannot be automated. Bot state is synchronized in the production database, so render jobs also have been set up to update the database when renders complete or error.

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.

ACM 978-1-4503-2261-4/13/07

Automated processes include detecting when shots are complete in upstream departments with no fixes open, performing QA steps on the shot, rendering bless frames and confirming that they match frames from lighting, rendering the full shot for both mono and stereo, and loading completed renders for a final manual render check by the rendering team. QA steps include the following: checking for fixes opened after the shot has entered rendering, checking frames for errors (such as bad pixel values, image format, or resolution), checking that render stats fit within specified gates, checking for shot files that are not installed, confirming that all render layers are used in the composite and all layers expected by the composite are rendered in the shot, and comparing blessed frames from lighting to the final rendered frames.

The interactive chat bot serves to redistribute shots to TDs when a QA step fails or render errors. When a shot cannot be handled by the automatic pipeline, the bot requests assistance in chat and assigns shots to TDs who volunteer for work. TDs can volunteer to take shots using natural language, mirroring workflows they used on previous productions to get work from a human coordinator.

The chat bot also gives live updates to TDs on their shot statuses, for example, when a TD's shot is done rendering or has errored. Production management gets notifications when consistency problems are detected, such as if multiple renders of a single shot are being run simultaneously. To prevent overcommunication, the bot keeps track of who it has notified of what, and only reminds a user of a problem that needs attention at longer intervals after the initial alert. Multiple alerts are consolidated into a single message–for example, multiple TDs get alerts that shots are ready for work in a single message rather than one message per TD or shot.

To prevent ready shots with problems from being missed over a weekend–say a shot is marked ready for rendering at the end of day on Friday, but there are still files checked out in upstream departments–the bot will note that the shot has potential problems and render it over the weekend anyway, reassigning the shot to a TD on Monday to investigate whether the frames can be used or not.

(03:11:33 PM) **lurch:** akolliop, ariela, dschmidt, senn: these shots are blessdone/error/farmdone in rendering and stereo_rendering: s510_9a, s610_23b, s610_40a, s645_11, s665_105, s690_29c
(03:13:22 PM) lurch: I have 3 shots that need help, can anyone take one?
(03:13:44 PM) krissy: ariela, s510_26p is ready for you! (this one has a fix attached)
(03:13:50 PM) senn: lurch shot
(03:13:50 PM) lurch: senn, s343_9m is yours because an initial QA step failed
(03:13:50 PM) lurch: 2 shots remain, if anyone can take one
(03:14:27 PM) senn: lurch give me another one
(03:14:27 PM) lurch: senn, s645_28b is yours because an initial QA step failed
(03:14:27 PM) lurch: 1 shots remain, if anyone can take one
(03:15:45 PM) marlena: lurch shot
(03:15:45 PM) lurch: marlena, s645_37 is yours because an initial QA step failed
(03:15:45 PM) lurch: All shots have been assigned, thanks!

Figure 2: *A typical interaction with lurch.*

At the point of this writing in the production of *Monsters University* with 50% of the film approved, 38% of the finaled shots have been processed by lurch. This has significantly reduced the number of shots TDs need to process, reducing workload and distractions.