The Advanced Rendering Toolkit (ART)

February 2005

Overview

ART is a highly modular software system for the synthesis and manipulation of photorealistic images which aims to incorporate a large number of cutting-edge technologies into an integrated, stand-alone toolkit.

The dual goal of this effort is to both serve as a software platform for research in the area of artistic and predictive rendering, and also to eventually yield state-of-the-art results as a production rendering system.

Its origins lie in the research efforts of the Institute of Computer Graphics and Algorithms at the Vienna University of Technology, but it was always intended to eventually also be useable outside the realm of academia. The project is an open source effort governed by the GNU Public License for Libraries (LGPL).

Until some time ago, it was possible to write an entire rendering system which conformed to the state of the art from scratch just for one's PhD – and discard it afterwards.

The tremendous advances of the recent years mean that this is no longer feasible, and that meaningful photorealistic rendering research needs a stable but sophisticated software platform on which one can base work that truly advances the field.

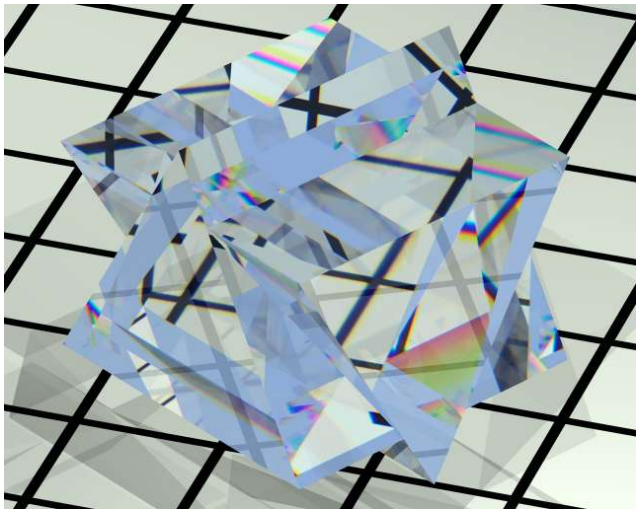
ART is about providing just such a platform.

System Particulars

ART is a collection of currently 22 small to medium-sized libraries written in a mix of ANSI C99 and Objective C with gnu99 extensions enabled, and a small number of applications which use these libraries. The percentage of the two languages in the project is not even: only those few core modules for which speed is essential are written in ANSI C99, and practically all high-level APIs are coded in Objective C.

The design of the whole toolkit – and in particular the notion of coding performance-critical modules in plain C while expressing high-level concepts in the Smalltalk-like semantics of Objective C – was strongly influenced by the philosophy of the NeXTStep computing environment on which development started in 1996; since then development of the project has continued on a variety of platforms, mainly Apple OS X and Linux.

While the somewhat exotic choice of languages might seem a bit odd at first, the clear and expressive semantics of Objective C (which is very easy to learn due to its simplicity) and the longstanding and rock-solid support of Objective C by the Gnu `gcc` compiler yield such a stable and productive working environment that we would make the same choice again if we were to start the project from scratch today.



Main Purpose and Workflow

ART is a toolkit whose primary function is to generate static high-quality images through advanced ray-based stochastic non-realtime rendering algorithms, similar to modern versions of POVray or Radiance, or commercial tools like Brazil® and mental ray®. It mainly uses a proprietary – but of course documented – scene description language, and offers a choice of several rendering algorithms to calculate intermediate high-dynamic range images which are stored in lossless, proprietary formats.

These proprietary image formats are also completely documented, and were made necessary by the fact that ART can generate and manipulate spectral HDR images – which optionally can even contain polarisation information – and no existing format was capable of storing such information.

The images generated by the rendering engine can then be processed by the provided tone mapping techniques and ultimately converted to displayable conventional image formats.

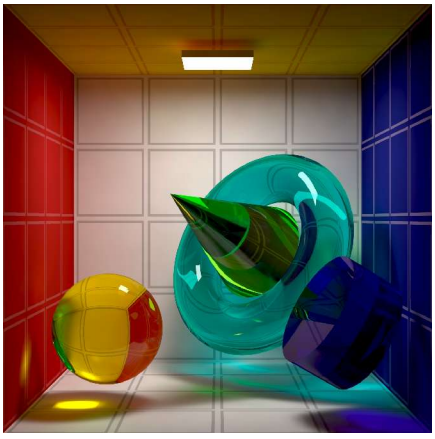
Apart from the core rendering functionality several necessary specialised utilities which are not available anywhere else – such e.g. a difference image generator for spectral images, or a falsecolour display for polarised images – are provided.

Features

Arguably rendering toolkits are – or at least used to be – a very common research project at universities, and commercial stochastic high-quality rendering systems such as Splutterfish's Brazil renderer have been available for some time now. What are the features of ART that currently make it a unique project in the graphics world?

In no particular order some of these are:

- **ART uses a highly flexible stack-machine paradigm to describe its actions.** Rendering jobs are specified as a combination of scene geometry, camera and the action sequence which is to be performed on this scene rather than using a static approach where rendering parameters can only be altered from the command line. The action sequence typically contains things such as the insertion of the acceleration structures, the actual rendering as well as the tone reproduction and gamut mapping actions, so that in essence the entire photorealistic rendering pipeline can be stored in the ART scene file. This feature is extremely handy both for eventual production use as well as for experimental work.



- **ART is a spectral rendering system.** It can use spectral representations of light and reflectancies for all its computations, and is therefore capable of predicting the appearance of scenes if measurements of the involved surface reflectancies and lights are available. However, ART can also be compiled to perform its computations in colour space if this is desired.
  The need to handle spectral quantities of radiance and reflectivity were a key reason to develop our own scene description language, since no existing SDL offers such capabilities. However, the parser subsystem was written with extensibiltiy in mind, and incorporating a different format – such as XML3D – would not require significant effort.
- **ART can compute the appearance of fluorescent materials.** As an extension to its spectral rendering capabilities, ART can also handle re-radiation of incident energy on different wavelengths. This feature is potentially highly valuable for the prediction of object appearance since a large number of modern dyes and pigments are fluorescent to some degree in order to increase the purity and intensity of the resulting colour.
- **ART can take the polarization state of light into account** during its calculations. This feature is rarely needed, but can be very important for predictive rendering in certain circumstances, e.g. when crystals, mirrors, glass panes or similar specular objects are involved, and the highest possible accuracy is desired for the interreflections between such objects.
- **ART offers a large number of options for describing shape.** ART is capable of directly using CSG, polygonal, NURBS and subdivision surface object descriptions, as well as natively supporting grammar-based modelling through L-systems. ART is fully on par with commercial toolkits in this respect, and exceeds the functionality of many such systems by offering a large number of different modelling paradigms side-by-side.
- **ART has a fully fledged shader language.** ART incorporates a genuine shading language which is only slightly less powerful than the RenderMan specification. And the majority of these differences to the PRMan featureset is due to constraints that only arise from using a shading language in a physically based renderer.

Development Work

ART is currently still under active development at the Institute. We use a central CVS repository as a means of coordinating the work of the (on average) ~5 active collaborators, who are made up of both staff members and students.

Status of the Toolkit

As of February 2005, the toolkit is in a feature-complete but still partially unfinished state, which means that the components are all in place and mostly functional, but that not all of them are finished and fully documented.

As a consequence, the main focus of our work is currently to further stabilise the existing functionality, and to only significantly extend it once a fully stable and well documented state has been reached. This stable version will also be the first to be made publicly available.

Due to the fact that development of production code (as opposed to the proof-of-concept implementations usually associated with research work) is not a priority at the university, an exact schedule for this first public release is currently impossible to give.

Contact & Further Information

Currently, the coordinator of ART development at the Institute is Alexander Wilkie; he can be reached at

mailto:wilkie@cg.tuwien.ac.at

The project website at http://www.artoolkit.org/ is currently out of date, but part of the release effort will be to re-launch it and provide detailed information on the status and scope of the project.