A scalable framework for image-based material representations

Tobias Franke, Dieter W. Fellner
Fraunhofer IGD
tobias.franke@igd.fraunhofer.de
Introduction

Overview

- Faithful digital reproduction of real artifacts
  - E.g. online CH galleries
  - "Digital Replica"
- Key ingredient: real world, measured material functions
  - No support in X3D
  - CSS and standard materials don’t address this
- Usual option: write your own shader, not portable
Introduction
Problem definition

- 3D-Coform
  - Cultural Heritage gallery based on X3DOM
  - All materials are rendered with the standard X3D material

3D Data Catalogue
http://3dcoform.eu/x3domCatalogue/
Introduction

Problem definition

3D Data Catalogue
http://3dcoform.eu/x3domCatalogue/
Introduction

Problem definition

3D-COFORM
Tools & Expertise for 3D Collection Formation

Flask
- Place of origin: Moustiers-Sainte Marie (made)
- Date: 19th century (made)
- Artist/Maker: Olérys and Laugier's pottery factory (possibly, manufacturer)
- Materials/Technique: Tin-glazed earthenware, painted in colours
- Museum number: 130-1930
- Gallery Location: Ceramics Study Galleries, Asia & Europe
Introduction

Problem definition

- InscriptiFact
- Archive for near eastern and mediterranean inscriptions
- PTM based
- Java viewer
- 2D

InscriptiFact
http://www.inscriptifact.com
Introduction
Problem definition

- HP Antikythera Mechanism
- Online gallery
- PTM based
- Java viewer
- 2D

Interactive Relighting of the Antikythera Mechanism
http://www.hpl.hp.com/research/ptm/antikythera_mechanism/
Introduction

Related Work

- BTFs
  - Approximated BTFs
  - WebGL streaming implementation
- Polynomial Texture Maps
- Precomputed Radiance Transfer
- CommonSurfaceShader

WebGL-based Streaming and Presentation Framework for Bidirectional Texture Functions
Case studies
Approximated BTFs

- Stack of textures
  - only one varying incident illumination angle \( \theta \)
  - only one view
  - usually \(~15\) images

- At runtime
  - interpolate texel from stack according to incident lighting angle/lambert factor

BTF Database Bonn and Measurement Lab
http://cg.cs.uni-bonn.de/en/projects/btfdbb/
Case studies
Polynomial Texture Maps

- Specialized texture format
  - 6 coefficients per pixel (per color channel)
- One view, varying incident illumination
- At runtime
  - project lighting angle into texture space, evaluate polynomial

HP Labs
https://www.hpl.hp.com/research/ptm/
Case studies
Sum of Gaussians

- Two textures encoding BRDF as sum of Gaussians
  - 1 texture for RGB weights
  - 1 texture for mean and standard deviation
- At runtime
  - Multiply with Gaussian-prefiltered environment map to calculate final value
Case studies
Precomputed Radiance Transfer

- Textures transmit coefficients
  - Each texel has $n^2$ coefficients
  - Divide up to RGB/RGBA textures
- At runtime
  - Multiply coefficients at each texel with distant environment lighting coefficient

Microsoft DirectX SDK
Case studies
Similarities

- Image-based materials share common features
  - 1..N textures encoding material property/transfer
  - Preprocessing & shader program for interpretation

- Usual traits
  - BRDF required for indexing
  - May be affected by texture transformations
  - May support/require relighting through environment map
Implementation

ImageMaterial node

ImageMaterial : X3DAppearanceNode
{
    MFNode [in,out] materialImages NULL
    MFString [in,out] materialTags NULL
    SFString [in,out] compression "approxBTF"
    SFNode [in,out] environmentTexture NULL
    SFString [in,out] mapMode "auto"
    SFBool [in] useRegularMaterial FALSE
    MFFloat [in] scale []
    MFFloat [in] bias []

    ...
}

PolynomialTextureMap : X3DTextureNode
{
    SFNode   [in,out]  textureProperties  NULL
    MFString [in,out]  url               NULL
    SFImage  [out]     image_changed
    MFImage  [out]     coefficientsHigh_changed
    MFImage  [out]     coefficientsLow_changed
    SFString [out]     format_changed
}
Implementation
Usage example

<X3D xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' ...>
<Scene>
  <Shape>
    <ImageMaterial compression='gaussian' materialTags='"weights", "meanStdDev"'>
      <ImageTexture2D
        containerField='environmentTexture'
        url='env.png'/>

      <ImageTexture2D
        containerField='materialImages'
        url='weights.png' />

      <ImageTexture2D
        containerField='materialImages'
        url='meandev.png' />
    </ImageMaterial>

    <Teapot />
  </Shape>
</Scene>
</X3D>
Discussion

Flexibility vs. self-descriptive interface

- Many other image-based schemes can be supported with the ImageMaterial rapidly.

- Major point of critique: brdfImages field too broad
  - Do-it-all interface not self-descriptive
  - Distribution of responsibility not good
  - Can’t parameterize!

- Possible solution: encapsulate special textures for material description in texture nodes.
Discussion

Flexibility vs. self-descriptive interface

ImageMaterial

Shape

Texture 1  Texture 2  Texture 3  Texture 4

“weights”, “colors”, “more weights”, “stuff”
Discussion
Alternative 1

- Solves
  - Parameterization at texture container level
  - Texture containers can provide specialized input slots for images
- Requires
  - Container for each image-based material
- Additionally solves flexibility when handling multi-purpose descriptors
  - E.g. Sum of Gaussians can be combined with PRT
Examples
Sum of Gaussians without/with visibility term
Examples
Sum of Gaussians NPR-like appearance

Mean & StdDev

Weights
Examples
Polynomial Texture Mapping
Examples
Approximated Translucency
Conclusion
... and Future Work

- Flexible solution to incorporate upcoming image-based methods to represent materials
- Need more experience if custom render styles are needed or texture containers can provide enough information
- Add more rendering methods
  - Abundance of PRT methods
  - Other BTF compression schemes, i.e. WebGL implementation of Klein et. al
Thanks for listening.

PTMConvert Code:
http://code.google.com/p/ptmconvert/
Questions?
tobias.franke@igd.fraunhofer.de