Tutorial

Bidirectional Texture Functions Modelling

Modelling & Editing

Michal Haindl Jiří Filip

Institute of Information Theory and Automation
of the AS CR

13/06/2010
CVPR 2010

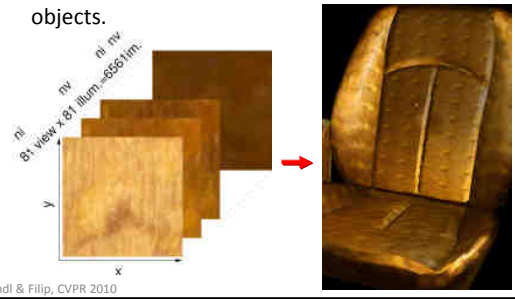
www.utia.cz

Outline

- Motivation
- Parametric Models
- Sampling Methods
  - Tiling, Roller
- Reflectance Models
- Markov Random Models
  - Smooth Texture Model
  - Probabilistic BTF Model
  - Compound Model
- BRDF Texels
- Mixture Models
- BTF Editing

Motivation

- BTF ⇨ image sequence ⇨ limited spatial resolution ⇨ disturbing repetitive effects when mapped on large objects.



Motivation
Image Modelling

- general model
  - image analysis
    - (un/semi)supervised segmentation, retrieval
    - inhomogeneity (defect/edge) detection
    - motion detection
  - image synthesis
    - static / dynamic texture enlargement
    - editing
    - restoration
    - missing data reconstruction (film scratches)
    - image compression (storage, transmission)
  - description of arbitrary complex spatial relationship
  - a general multivariate distribution simulation

Motivation
Image Modelling

- image modeling is extremely difficult
  - **Still image variability**
    Huge number of images $2^{n\,N\,M\,l}$, e.g.
    $n = 8,\ N = M = 512,\ l = 3 \longrightarrow \sim 10^{1.893.917}$
  - **Dimensionality**
    4**D** – video, **7D** – still BTF, **8D** – dynamic BTF
  - **Large still images**, e.g., a Landsat scene:
    185×185 km $(2800 \times 2800$pixels$)$, 7 spectral bands, 8 bits/pixel, 460 Mbit
  - **4D images (dynamic)**, e.g. ~ 1GB a movie in TV quality

Motivation
1-Dimension Versus n-Dimensions

- n-D non trivial generalization of 1-D random processes
  - ☒ Causality:
    1-D natural
    n-D artificial (no natural order)
  - ☒ No n-D factorization theorem (general 2D random field (RF) no unilateral representation)
  - ☒ 1D techniques often intractable in n-D
- 2D ideas ⇨ better results than 1D ⇨ 2D

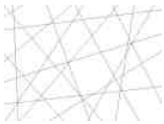
Texture Parametric Models

- Random Mosaic Models
- Periodic Tesselated Models
- Syntactic Models
- Fractal Models
- Second-Order Statistical Models
- ARMA, MA
- Reflectance Models
- Mixture Models
- Markov Random Fields (MRF)

Texture Parametric Models
Random Mosaic Models

- **Synthesis**
  - Region tesselation into cells
  - Cell class assignment $P_1, \ldots, P_K$
- **Analysis**
  - Variogram

$$\begin{aligned} V(d) &= E\{(Y_r - Y_{r+d})^2\} \\ V(d) &= 2\sigma^2(1 - P(d)) \\ P(d) &= P(Y_r = \omega, Y_{r+d} = \omega) \end{aligned}$$



  - Poisson line model

Texture Parametric Models
Fractals

- a deterministic fractal - a bounded set S in a Euclidean n-space is **self-similar** is the union of distinct (non-overlapping) copies of itself, each has been scaled down by a ratio *r*.
- fractal dimension estimation [Voss 71]:

$$E\{N(L)\} = M \sum_{m=1}^{N} \frac{1}{m} P(m, L) \propto L^{-D}$$

- $P(m, L)$- probability that *m* points of an image surface are within a box of side length *L*, *M* number of points in image.

☒ natural textures requires stochastic fractals ⇨ difficult fractal dimension estimation

☒ two very different textures can have similar fractal dimension

☒ fractal synthesis - limited textures

☑ scalability

Markov Random Fields

- **Random Field** (RF) $\{Y_r\} r \in I$ countable (no continuous) index set
- RF with { denumerable | real } state space ω
- Ω the space of all configuration from ω
- p probability measure on Ω such that

$$p(Y_r;\ r \in A) > 0 \qquad \forall A \subset I$$

- $\mathcal{B}$ minimal complete σ-algebra
- RF: $(\Omega, \mathcal{B}, p, \{Y_r\})$
- **Markov Random Field** (MRF) + Markovianity condition:

$$p(Y_r \,|\, Y_s\ \forall s \in I \setminus \{r\}) = p(Y_r \,|\, Y_s\ \forall s \in I_r)$$

Markov Random Fields

☑ generality - any nD RF is a MRF $I_r = I \setminus \{r\}$

☑ Consistent global and local representation

☑ regular, irregular (graph), finite, infinite index sets

☑ Bayesian formulation of image processing tasks

☑ simpler MRF models can be combined (compound MRF)

☑ Global optimization tool

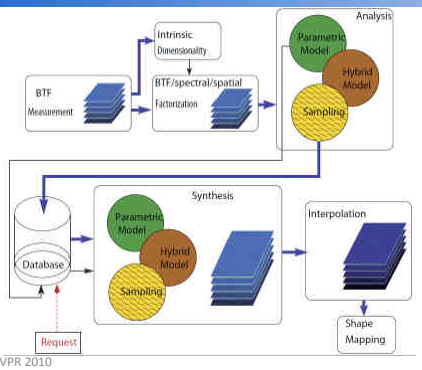
☒ Several open problems

☒ Demanding Markov chain Monte Carlo methods

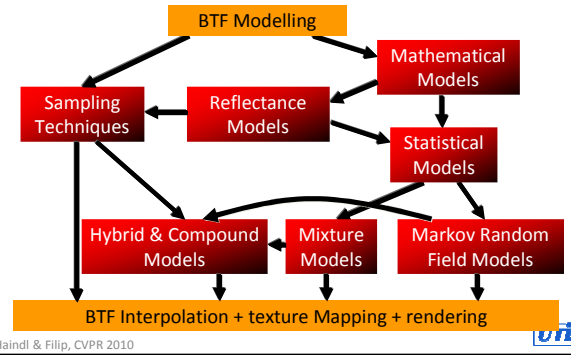
Markov Random Fields
Multidimensional Random Fields

- 3D static multispectral image,
- 4D colour video (dynamic texture),
- 7D static BTF texture,
- 8D dynamic BTF texture.
- nD models
  - 3D
- approximations of nD model with $n \times (n-1)D$ models
  - ☒ approximation error
- fixing some indices
  - ☒ modeling restriction, huge parametric space

BTF Modelling
Generic BTF Model



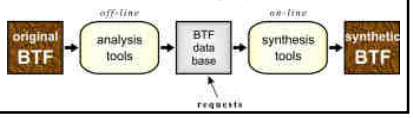
BTF Modelling Taxonomy



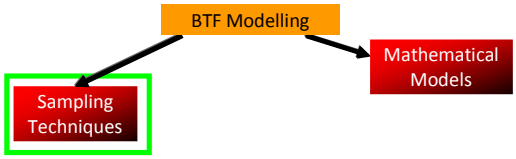
BTF Modelling

- Sampling-based approaches
  - Texture Tiling and patching
  - Roller
- Model-based approaches
  - Reflectance Models
  - MRF-based models
    - Multiscale 2D / 3D causal autoregressive
    - Multiscale 2D / 3D Gaussian MRF
  - Mixture Models
    - Discrete / Gaussian / Bernoulli

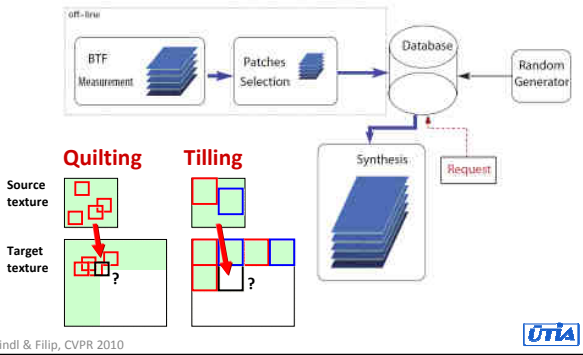
Modelling pipeline



BTF Sampling

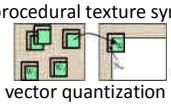


BTF Sampling

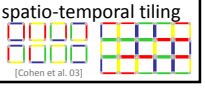


BTF Sampling
Image Sampling Approaches

- [Heeger & Bergen 95] – pyramid-based texture analysis & synthesis
- [De Bonet 97] – samples from conditional distribution over multiple scales
- [Xu et al. 00] – Chaos mosaic: fast procedural texture synthesis
- [Efros 01] – Patch-based synthesis
  ⇨ Image Quilting

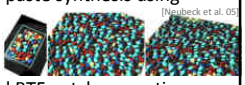

[Efros 01]

- [Wei & Levoy 01] – Tree-structured vector quantization constrained by a local geometry
- [Ashikhmin 01] – … modification by irregular regions copying
- [Zelinka & Garland 03] – Fast synthesis using jump-maps
- [Kwatra et al. 03] GRAPHCUT ⇨ universal spatio-temporal tiling
- [Cohen et al. 03] – Wang tiles ⇨ aligned to match edge color


[Cohen et al. 03]

BTF Sampling
Approaches Overview

- [Liu et al. 01] – combining a sparse set of BTF measurements according to quilting-based synthesis of material range-map
  - ☒ no analysis-synthesis separation ⇨ slow synthesis, visible seams
- [Tong et al. 02] – BTF synthesis based on similar surface textons ⇨ BTF pixels are assigned texton labels
  - ☒ Slow synthesis
- [Neubeck et al. 04] – iterative copy & paste synthesis using [Ashikhmin 01] candidate search.
  - ☒ No separation of analysis and synthesis
  - ☒ Small size neighborhoods restriction


[Neubeck et al. 05]

- [Zhou et al. 05] – image quilting-based BTF patches creation and interactive editing
  - ☒ Very slow

[Zhou et al. 05]

BTF Sampling
Approaches Overview

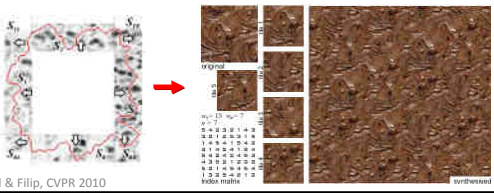
- [Kawasaki et al. 05] – image-quilting of PCA-compressed spherical harmonics BTF expansion
  - ☒ no analysis-synthesis separation ⇨ slow synthesis
- [Leung et al. 07] – BTF data as Wang tiles in spherical harmonics domain
  - ☑ Real-time rendering on arbitrary surface,
  - ☑ interactive tiles editing

[Leung et al. 07]

- [Somol&Haindl 05] - image tiling by a set of arbitrarily connectable tiles (the same boundary, different content)
  - ☑ Real-time BTF rendering
- [Haindl&Hatka 05] - tiling based on double toroidal patches
  - ☑ Real-time rendering
  - ☑ Fast tiles generation

BTF Sampling
Tiling

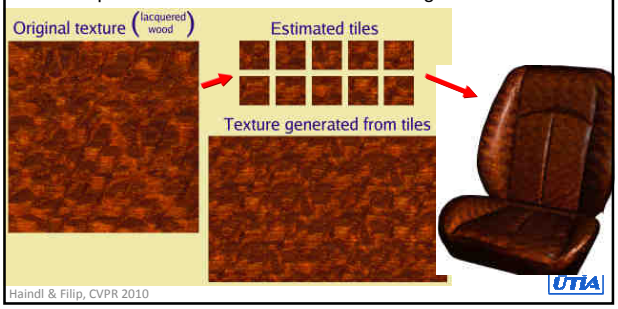
[Somol&Haindl 05] – looking for sub-optimal path in error-map of overlapped texture regions + changing tile contents



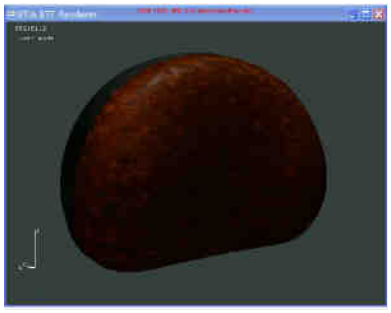
BTF Sampling
Tiling

Example of BTF tiling [Somol&Haindl 05]

Find optimal cut of tiles across all BTF images

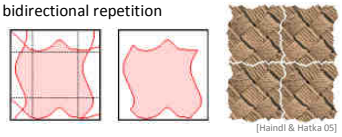


BTF Sampling
Tiling Demo



BTF Sampling
Roller

- **Principle:** double toroidal tile [Haindl &Hatka 05]
- **Algorithm:**
  - **Analysis**
    1. Find the optimal horizontal and vertical overlaps h^*, v^*.
    2. Find the optimal vertex r^* .
    3. Search for optimal horizontal and vertical cuts starting from r^* with final points in the corresponding rectangle vertices.
    4. Create the double toroidal texture tile.
  - **Synthesis** – bidirectional repetition


[Haindl & Hatka 05]

BTF Sampling
Roller

- Tile – optimal overlap

$$h^* = \min_h \left\{ \frac{1}{h} \sum_{\forall r \in I_h} \psi_r^h \right\} \qquad \psi_r^h = \left(Y_r - Y_{r+[N-h,0]}\right)^2 \quad \forall r \in I_h \ ,$$

$$v^* = \min_v \left\{ \frac{1}{v} \sum_{\forall r \in I_v} \psi_r^v \right\} \qquad \psi_r^v = \left(Y_r - Y_{r+[0,M-v]}\right)^2 \quad \forall r \in I_v$$

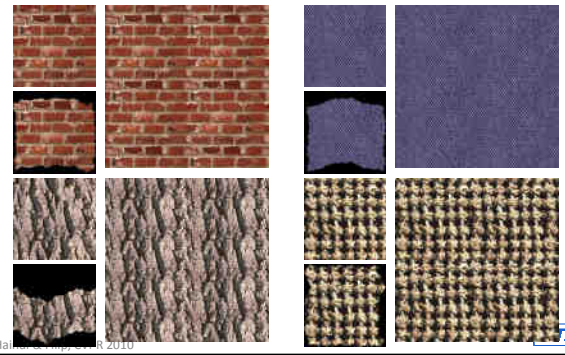
$$I_h = (1,\ldots,h) \times (1,\ldots,M) \ , \qquad I_v = (1,\ldots,N) \times (1,\ldots,v) \ ,$$

- Tile – optimal cut
  - dynamic programming (fixed starting points)

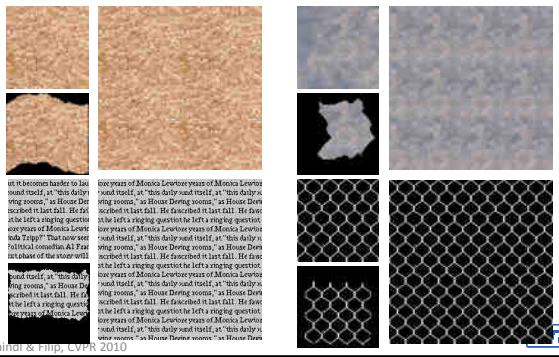
$$\Psi_r^{h^*} = \psi_r^{h^*} + \min\left\{\Psi_{r-[1,1]}^{h^*}, \Psi_{r-[0,1]}^{h^*}, \Psi_{r+[1,-1]}^{h^*}\right\}$$

$$\Psi_r^{v^*} = \psi_r^{v^*} + \min\left\{\Psi_{r-[1,1]}^{v^*}, \Psi_{r-[1,0]}^{v^*}, \Psi_{r+[-1,1]}^{v^*}\right\}$$

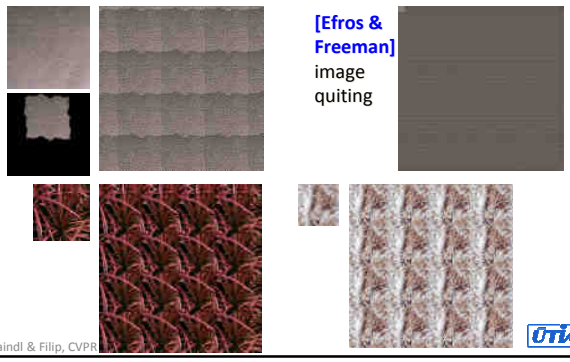
BTF Sampling
Roller - examples



BTF Sampling
Roller - examples



BTF Sampling
Roller - limits



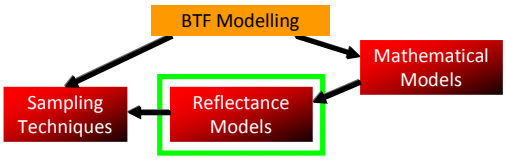
[Efros & Freeman] image quiting

BTF Sampling

Summary

- ☑ High visual quality
- ☑ Often numerically simple (GPU)
- ☒ Low compression
- ☒ Observable repetitions (low frequencies, large textures)
- ☒ Unseen textures / view / illumination angles impossible

BTF Reflectance Models



BTF Reflectance Model

Lafortune BRDF model [Lafortune97]

- physically plausible reflectance model with n_l lobes

$$Y_v(i,k) = \sum_{k=1}^{n_l} \rho_{v,k}(C_{X,v,k}u_x + C_{Y,v,k}u_y + C_{Z,v,k}u_z)^{n_{v,k}}$$

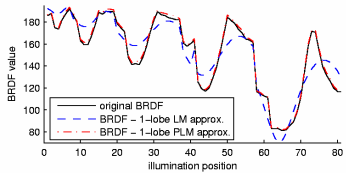
- one-lobe variant applied to separate view

$$Y_v(i) = \rho_v(C_{X,v}u_x + C_{Y,v}u_y + C_{Z,v}u_z)^{n_v}$$

  - only 5 unknown parameters ρ, C_X, C_Y, C_Z, n for each RGB spectral band
  - easy fitting ⇨ non-linear parameters estimation
- **81** BTF images represented using **5** parametric images
- each pixel ⇨ dedicated Lafortune model

BTF Reflectance Model

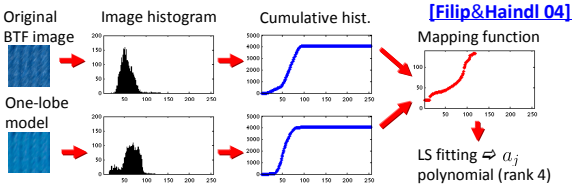
- Pixel reflectance during changing illumination position (view position fixed) – material **wool**



— Original reflectance shape
— One-lobe model approximation
— One-lobe model approximation + polynomial extension

BTF Reflectance Model
Parameters Fitting

[Filip&Haindl 04]



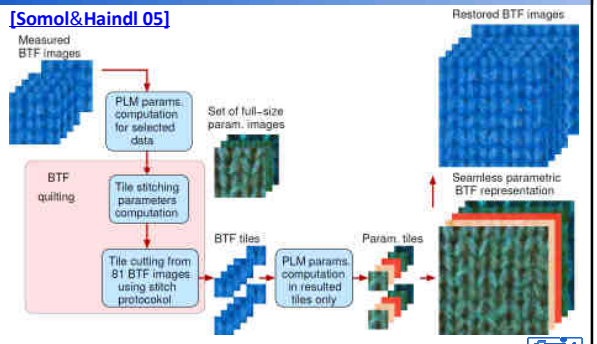
- Resulted model ⇨ polynomial expansion of one-lobe Lafortune model using coefficients $a_j, j = 0 \ldots 4$

$$\hat{Y}_v(i) = \sum_{j=0}^{4} a_{v,i,j} Y_v(i)^j = \sum_{j=0}^{4} a_{v,i,j}[\rho_v(C_{X,v}u_x + C_{Y,v}u_y + C_{Z,v}u_z)^{n_v}]^j$$

- Each BTF image ⇨ **15** additional coefficient to store

BTF Reflectance Model
Parameters Sampling

[Somol&Haindl 05]

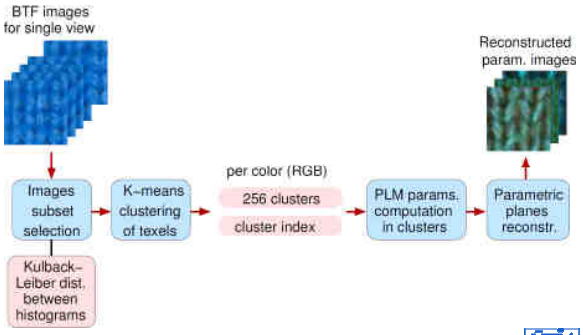


BTF Reflectance Model
Parameters Clustering

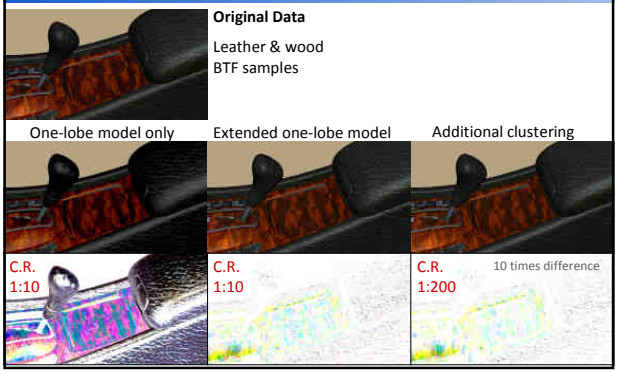
- Proposed model compression ~**1:10** ⇨ too low
- K-Means clustering of parametric tiles
- Achieved BTF compression **~1:200**

[Filip&Haindl 05]

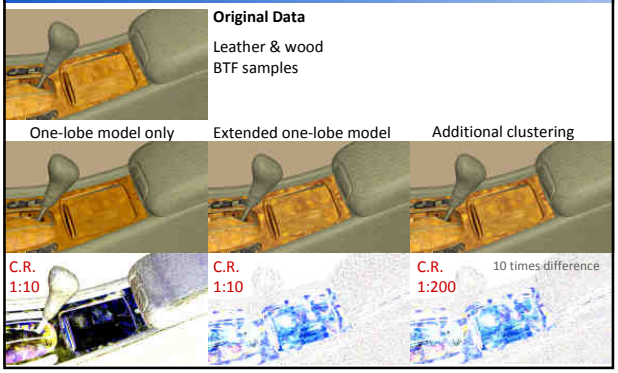
BTF Reflectance Model
Parameters Clustering



BTF Reflectance Model
Results



BTF Reflectance Model
Results



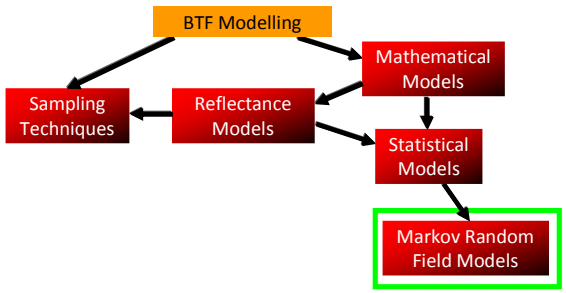
BTF Reflectance Model

Summary

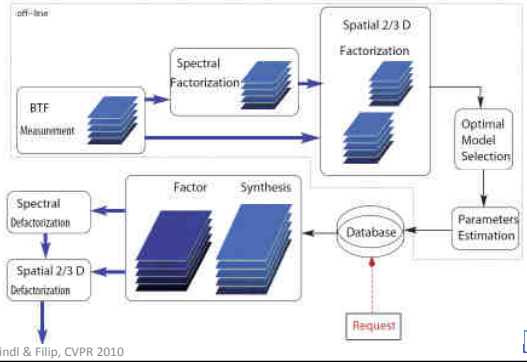
- ☑ High visual quality
- ☑ BTF seamless enlargement – parametric space sampling
- ☑ Fast GPU implementation possible

- ☒ Modest compression ratio $\approx 1:200$
- ☒ Initial values for parameter numerical estimation
- ☒ One-lobe is insufficient for some materials
- ☒ No unseen data

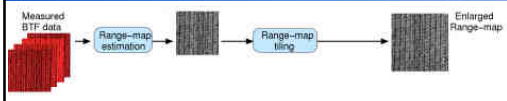
Probabilistic BTF Model



Probabilistic BTF Model



Probabilistic BTF Model

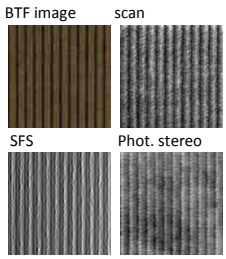


Range-map estimation

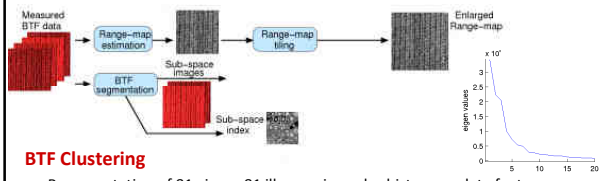
- **Direct measurement**
- **Shape from shading** [Frankot 88]
  - Coarse estimate from one image.
  - Assumption of Lambertian surface

$$Y_r = R(z_{r_1}, z_{r_2}, \theta, \mathrm{C}, \rho) = \rho \frac{\theta_1 z_{r_1} + \theta_2 z_{r_2} + \theta_3}{\sqrt{\theta_1{}^2 + \theta_2{}^2 + \theta_3{}^2}\sqrt{1 + z_{r_1}^2 + z_{r_2}^2}}$$

- **Over-determined Photometric stereo**
  - 81 ideally registered images

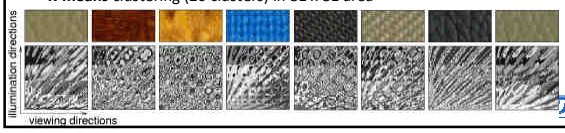


Probabilistic BTF Model

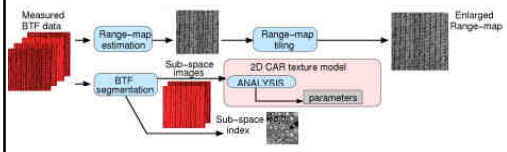


BTF Clustering

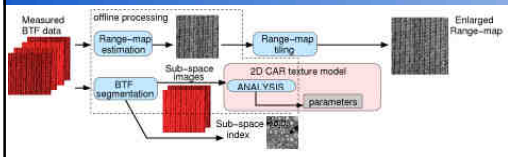
- Representation of 81 view x 81 illum. using color histogram data features
- 10-20 eigen-vectors contain 95% of information
- **K-means** clustering (20 clusters) in 81 x 81 area



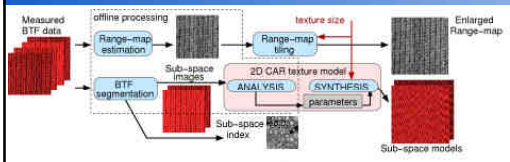
Probabilistic BTF Model



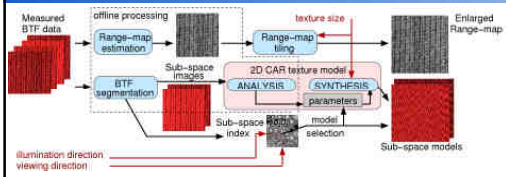
Probabilistic BTF Model



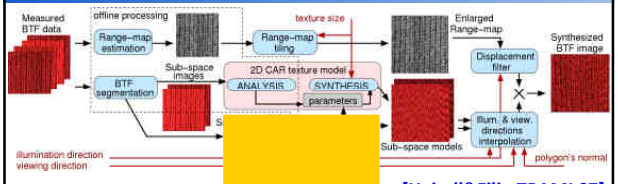
Probabilistic BTF Model



Probabilistic BTF Model



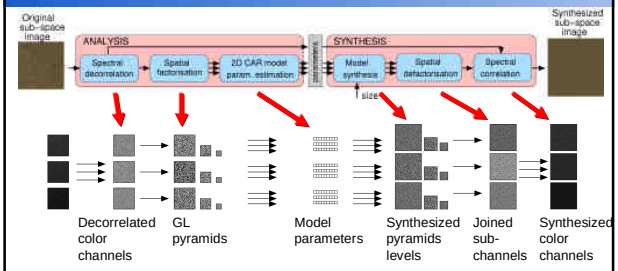
Probabilistic BTF Model



[Haindl&Filip TPAMI 07]

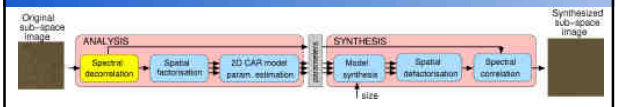
Smooth texture model…

Smooth 2D CAR Texture Model



Model's products

Smooth 2D CAR Texture Model

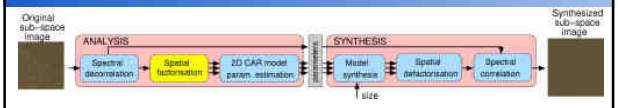


- **Spectral Decorrelation** using Karhunen-Loeve transformation

$$\bar{\mathbf{Y}}_{\mathbf{r},\bullet} = \mathbf{T}\tilde{\mathbf{Y}}_{\mathbf{r},\bullet}$$

where $\mathbf{T} = [u_1^T, u_2^T, u_3^T]^T$ is composed of eigen-vectors of matrix $\Phi = E\{\tilde{Y}_{r,\bullet}\tilde{Y}_{r,\bullet}^T\}$

Smooth 2D CAR Texture Model



- **Spatial Factorization** – decomposition of monospectral image into multi-resolution grid using Gaussian-Laplacian pyramid
  - Generation of Gaussian pyramid

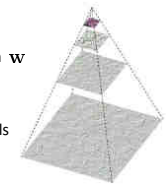
$$\ddot{\mathbf{Y}}_{\mathbf{r},\mathbf{i}}^{(\mathbf{k})} = \downarrow_{\mathbf{r}}^{\mathbf{n}} (\ddot{\mathbf{Y}}_{\bullet,\mathbf{i}}^{(\mathbf{k}-1)} \otimes \mathbf{w}) \quad \mathbf{k} = \mathbf{1}, \mathbf{2}, \ldots$$

using of convolution mask based on weight function $\mathbf{w}$

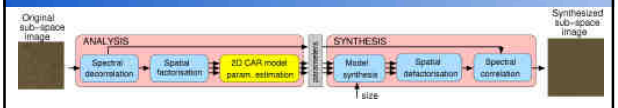
$$\ddot{\mathbf{Y}}_{\mathbf{r},\mathbf{i}}^{(\mathbf{k})} = \sum_{\mathbf{h},\mathbf{j}=-\mathbf{l}}^{\mathbf{l}} \hat{\mathbf{w}}_{\mathbf{h}}\hat{\mathbf{w}}_{\mathbf{j}}\ddot{\mathbf{Y}}_{\mathbf{2r}+(\mathbf{h},\mathbf{j}),\mathbf{i}}^{(\mathbf{k}-\mathbf{1})}$$

  - Laplacian pyramid from different Gaussian pyr. levels

$$\dot{\mathbf{Y}}_{\mathbf{r},\mathbf{i}}^{(\mathbf{k})} = \ddot{\mathbf{Y}}_{\mathbf{r},\mathbf{i}}^{(\mathbf{k})} - \uparrow_{\mathbf{r}}^{\mathbf{n}} (\ddot{\mathbf{Y}}_{\bullet,\mathbf{i}}^{(\mathbf{k}+\mathbf{1})})$$



Smooth 2D CAR Texture Model

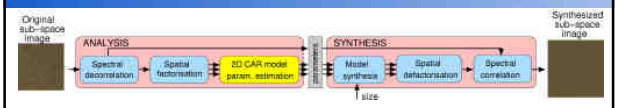


- **Support set estimation** – support set is required to be:

causal: I_r^c

unilateral: I_r^u

Smooth 2D CAR Texture Model



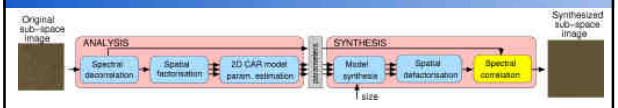
- **2D CAR model** can be expressed as a stationary causal uncorrelated Gaussian noise driven 2D autoregressive process:

$$\mathbf{Y}_{\mathbf{r}} = \gamma\mathbf{X}_{\mathbf{r}} + \mathbf{e}_{\mathbf{r}}$$

⇨ CAR parameters γ and σ^2 can be estimated analytically using Bayesian estimate (ML,LS).

$$\gamma_r = (\sum X_r X_r^T)^{-1}(\sum X_r Y_r^T)$$

Smooth 2D CAR Texture Model



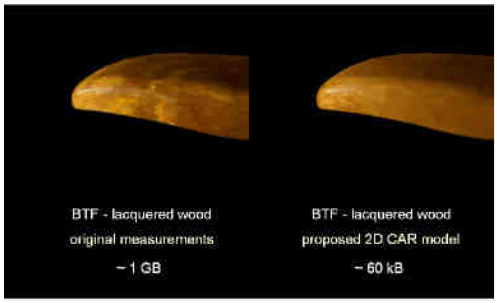
- **Model synthesis** $\mathbf{Y}_{\mathbf{r}} = \hat{\gamma}\mathbf{X}_{\mathbf{r}} + \mathbf{e}_{\mathbf{r}} \qquad \hat{\gamma} = \mathbf{E}\{\gamma|\mathbf{Y}^{(\mathbf{r})}\}$
- **Spatial defactorization** – Gaussian-Laplacian pyramid collapse $\longrightarrow \bar{\mathbf{Y}}_{\mathbf{r},\bullet}$
- **Spectral correlation** – inverse Karhunen-Loeve transformation $\tilde{\mathbf{Y}}_{\mathbf{r},\bullet} = \mathbf{T}^{-1}\bar{\mathbf{Y}}_{\mathbf{r},\bullet}$

Probabilistic BTF Model
Results 2D CAR – Smooth Samples

- Original Data Rendering – smooth samples - **2GB** data

- Results from model - **0.1MB** data

Probabilistic BTF Model
Demo



Probabilistic BTF Model
Results 2D CAR - Rough Samples

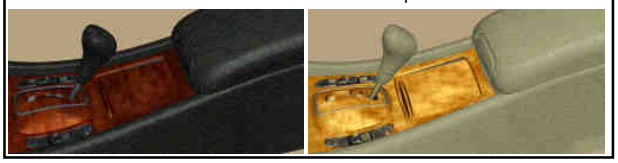
1. Original, 2. 2D CAR model, 3. relit geometry, 4. 2.+3. combined

1. 2GB
2.
3.
4. 0.3MB

Probabilistic BTF Model
Results 2D CAR

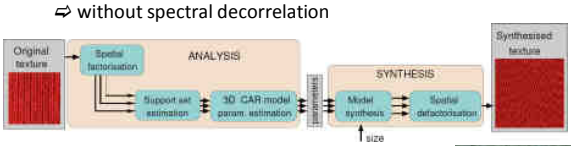
- Original Data Rendering ⇨ data 4GB

- Results from model ⇨ data 400 kB ⇨ compression ratio 1:10 000



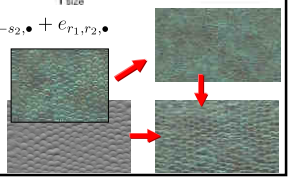
Alternative MRF Models
3D CAR Texture Model

- **3D Causal Auto-Regressive model** [Haindl&Filip 04]
  ⇨ without spectral decorrelation



$$Y_{r_1,r_2,\bullet} = \sum_{\{s_1,s_2\}\in I_{r_1,r_2}} A_{s_1,s_2} Y_{r_1-s_1,r_2-s_2,\bullet} + e_{r_1,r_2,\bullet}$$

$$e_{r_1,r_2,\bullet} \in \mathcal{N}(0,\Sigma)$$



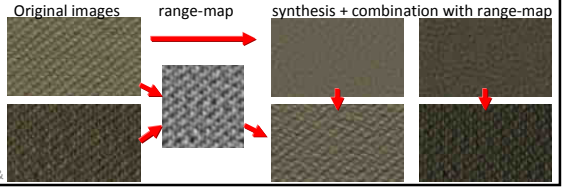
Alternative MRF Models
2D GMRF Texture Model

- **2D Gaussian Markov Random Field model** [Haindl&Filip 03]

$$\tilde{Y}_{r,i} = \sum_{s\in I_{r,i}} a_{s,i}\tilde{Y}_{r-s,i} + e_{r,i}$$

$$R_e(s,i) = E\{e_{r,i}\, e_{r-s,i}\} = \begin{cases} \sigma_i^2 & \text{if } s=(0,0) \\ -\sigma_i^2 a_{s,i} & \text{if } s \in I_{r,i} \\ 0 & \text{otherwise} \end{cases}$$

Original images range-map synthesis + combination with range-map



Alternative MRF Models
3D GMRF Texture Model

- **3D Gaussian Markov Random Field model**

$$\tilde{Y}_{r,\bullet} = \sum_{s \in I_r} A_s \tilde{Y}_{r-s,\bullet} + e_{r,\bullet}$$

$$R_e(s,i,j) = E\{e_{r,i}\, e_{r-s,j}\} = \begin{cases} \sigma_j^2 & \text{if } s = (0,0) \text{ and } i = j \\ -\sigma_j^2 a_{i,j}^s & \text{if } s \in I_r^{i,j} \\ 0 & \text{otherwise} \end{cases}$$

wood 3D GMRF 2D GMRF

Probabilistic BTF Model

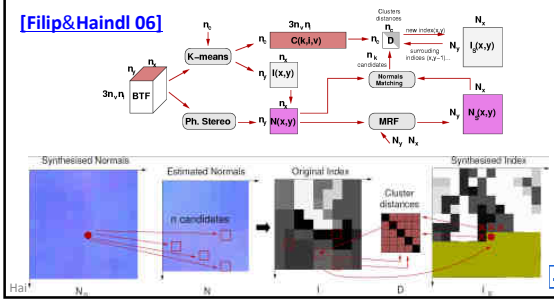
Summary

- ☑ Utmost compression ratio $\sim 1 : 10^5$
- ☑ Fast synthesis and analysis.
- ☑ BTF space restoration (from 1 image in extreme).
- ☑ Fast GPU implementation possible.
- ☑ Possible unseen texture modelling.
- ☒ Occasionally compromised visual quality.
- ☒ Stability test.
- ☒ Inappropriate for regular textures.
- ☒ General MRF problems.

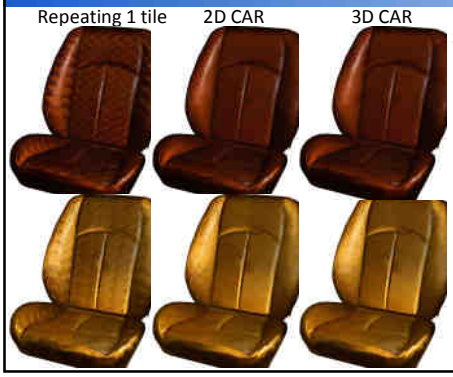
BRDF Texels

- Clustering ABRDF view/illum. Dependent texels using K-Means
- Enlargement of cluster index by 2D / 3D CAR models.

[Filip&Haindl 06]



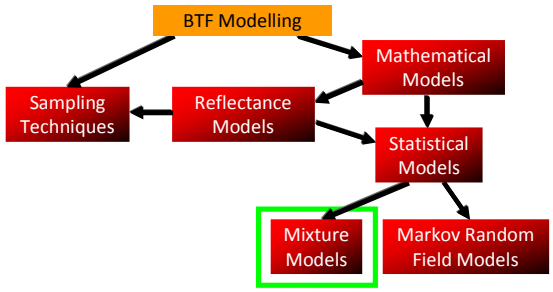
BRDF Texels



Summary

- ☑ Compression 1:500
- ☑ Reflectance behavior of clusters is untouched
- ☑ Fast rendering
- ☒ For smooth non-regular surfaces
- ☒ Slightly compromised quality

Mixture Models



Mixture Models
Probabilistic 2D Mixtures

- Spectral factorisation using PCA decorrelation.
- Monospectral / bit factor mixture models estimation.
- Texture synthesis by conditional distribution generators.
- Spectral defactorisation.

Factor mixture synthesis:

$$P(Y_{\{r\}}) = \sum_{m=1}^{M} P(Y_{\{r\}} \,|\, m)\, p(m)$$

$$Y_{\{r\}} = [Y_s \quad \forall s \in I_r \cup \{r\}]$$

Mixture Models
Probabilistic Discrete & Gaussian Mixtures

[Grim & Haindl 03]

$$P(Y_{\{r\}}) = \sum_{m\in\mathcal{M}} p(m)\, P(Y_{\{r\}}\,|\,m)$$

$$P(Y_{\{r\}}) = \sum_{m\in\mathcal{M}} p(m) \prod_{s\in I_r} p_s(Y_s\,|\,m)$$

Gaussian mixtures [Haindl et al. 04]

$$P(Y_{\{r\}}\,|\,m) = \prod_{n\in\mathcal{N}} \frac{1}{\sqrt{2\pi}\sigma_{mn}} \exp\left\{-\frac{(x_n-\mu_{mn})^2}{2\sigma_{mn}^2}\right\}$$

Mixture Models
Bernoulli Mixtures

[Haindl et al. 10]

$$\begin{aligned} p_s(Y_s\,|\,m) &= \theta_{m,s}^{Y_s}(1-\theta_{m,s})^{1-Y_s}\ ,\ Y_s\in\mathcal{B}=\{0,1\} \\ p_s(\cdot\,|\,m) &= (\theta_{m,s}, 1-\theta_{m,s}) \end{aligned}$$

Estimation using EM algorithm

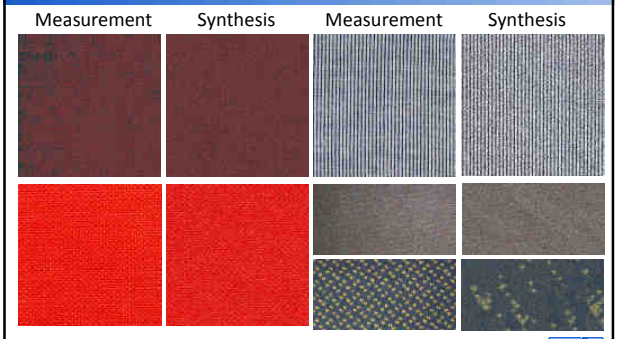
$$\begin{aligned} q^{(t)}(m|\,Y_{\{r\}}) &= \frac{P^{(t)}(Y_{\{r\}}\,|\,m)\,p^{(t)}(m)}{\sum_{j\in\mathcal{M}} P^{(t)}(Y_{\{r\}}\,|\,j)\,p^{(t)}(j)}, \\ p^{(t+1)}(m) &= \frac{1}{|\mathcal{S}|}\sum_{Y_{\{r\}}\in\mathcal{S}} q^{(t)}(m\,|\,Y_{\{r\}}), \\ p_n^{(t+1)}(\xi\,|\,m) &= \frac{1}{|\mathcal{S}|p^{(t+1)}(m)} \\ &\quad \sum_{Y_{\{r\}}\in\mathcal{S}} \delta(\xi, Y_n)\, q^{(t)}(m\,|\,Y_{\{r\}}), \quad \xi\in\mathcal{B} \end{aligned}$$

Mixture Models
Synthesis

Conditional probability density $Y_{\{\rho\}}$ known part of the window

$$\begin{aligned} p_{n\,|\,\rho}(Y_n\,|\,Y_{\{\rho\}}) &= \sum_{m=1}^{M} W_m(Y_{\{\rho\}}) p_n(Y_n\,|\,m) \\ W_m(Y_{\{\rho\}}) &= \frac{p(m)P_\rho(Y_{\{\rho\}}\,|\,m)}{\sum_{j=1}^{M} p(j)P_\rho(Y_{\{\rho\}}\,|\,j)} \\ P_\rho(Y_{\{\rho\}}\,|\,m) &= \prod_{n\in\rho} p_n(Y_n\,|\,m) \end{aligned}$$

Mixture Models
Bernoulli Mixtures - Results



Mixture Models
Bernoulli Mixtures - Results

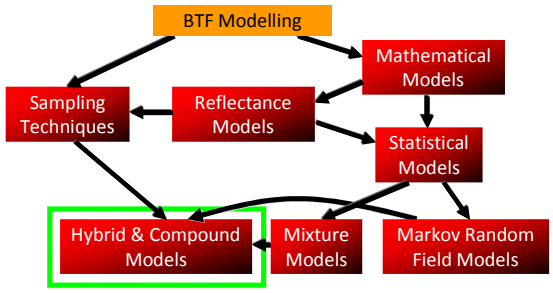
Synthetic gingham texture mapped on a snail shell model.

Mixture Models

Summary

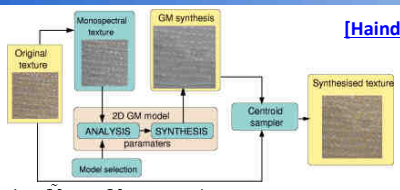
- ☑ Modest compression ratio $\sim 1:10^2$
- ☑ Allows semi-regular/regular textures modelling.
- ☑ Can be combined into more complex compound mixture/MRF models.
- ☒ Large training set (less robust than MRF).
- ☒ Time consuming parameter estimation.

Hybris & Compound Models



Hybrid BTF Models
Gaussian Mixture Model

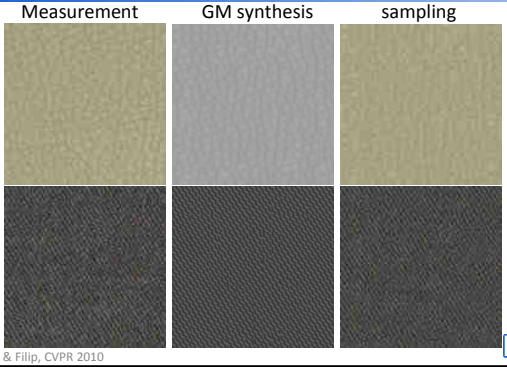
[Haindl et al. 05]



- BTF color $\tilde{Y} \longrightarrow Y$ gray-scale
- Gaussian Mixtures synthesis Y
- $\forall \mu_i$ find the most similar centroid:
$$Y_{\{r_i\}} = \arg\min_{Y_{\{r\}}} \{\|Y_{\{r\}} - \mu_i\|\}$$
- Optimal BTF sample from the original: $\mu_i \longrightarrow Y_{\{r_i\}} \longrightarrow \tilde{Y}_{\{r_i\}}$
- Boundary smoothing

Hybrid BTF Models
Gaussian Mixture Model

Measurement GM synthesis sampling



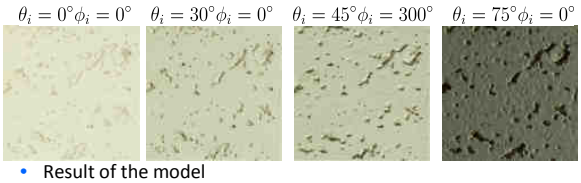
Compound MRF Texture Model

- Dedicated models in different structure regions:
$$P(X, Y \,|\, \tilde{Y}) = P(Y \,|\, X, \tilde{Y}) P(X \,|\, \tilde{Y})$$
$X_r \in \{1, 2, \ldots, K\}$ control RF, $\quad Y = \bigcup_{i=1}^{K} {}^{i}Y$ regional RF
- Maximal Aposterior Probability:
$$(\hat{X}, \hat{Y}) = \arg \max_{X \in \Omega_X, Y \in \Omega_Y} P(Y \,|\, X, \tilde{Y})\, P(X \,|\, \tilde{Y})$$
- Two step approximation: [Haindl&Havlíček10]
$$(\check{X}) = \arg \max_{X \in \Omega_X} P(X \,|\, \tilde{Y})$$
$$(\check{Y}) = \arg \max_{Y \in \Omega_Y} P(Y \,|\, \check{X}, \tilde{Y})$$

Compound MRF Texture Model

- Original BTF images (ceiling panel sample)

$\theta_i = 0^\circ \phi_i = 0^\circ$ $\theta_i = 30^\circ \phi_i = 0^\circ$ $\theta_i = 45^\circ \phi_i = 300^\circ$ $\theta_i = 75^\circ \phi_i = 0^\circ$



- Result of the model

BTF Hybrid & Compound Models

Summary

- ☑ High visual quality
- ☑ Fast synthesis (CMRF)
- ☑ High compression ration (CMRF), modest compression (hybrid)
- ☒ No analytical solution
- ☒ Part of original measurements stored

BTF Editing

- Automatic texture adjustment to the target texture.
- Expected visual output.
- Texture enlargement & huge compression.
- Fast synthesis & easy parallelisation.
- Interactive controls for users

BTF Editing

[Kautz et al. 07] - interactive editing of measured BTF

⇨ change materials properties by a set of physically non-plausible operators.

☒ After changes, recompression required

[Kautz et al. 07]

[Menzel & Guthe 09] – g-BRDFs – BTF parametrized into textures

⇨ interactive modification of textures modifies appearance.

☑ Very fast and intuitive

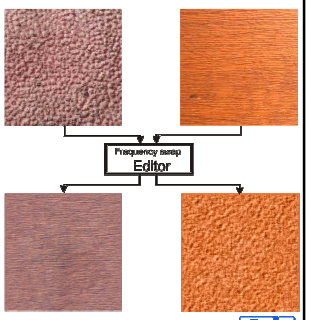
☒ Pixels are independent ABRDFs

⇨ no inter-reflection & sub-surface shadowing.

BTF Editing
Frequency Swap Strategy

[Haindl & Havlicek 09]

- Frequency factorization using the Gaussian-Laplacian pyramid.
- Markovian modeling of multispectral Gaussian-Lapacian pyramid layers using the 3D causal autoregression model (3DCAR).
- Synthetic (enlarged) Laplacian pyramid swapping between the input and target textures.



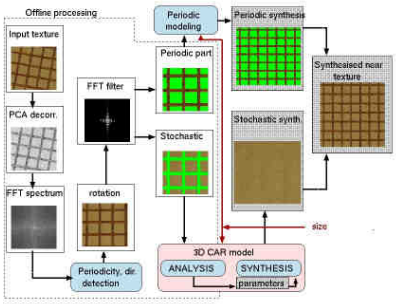
BTF Editing
Frequency Swap Strategy - Results

Synthesized textures:

(combinations of the previous)

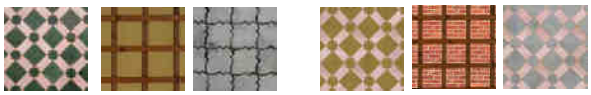
BTF Editing
Near Regular Textures

[Haindl &Hatka 09]



BTF Editing
Near Regular Textures

- Fully automatic and extremely fast method due to strict separation of the analytical and very efficient synthesis steps.
- Compression ratio > alternative tiling methods.
- Texture enlargement & editing.



BTF Editing

Summary

- ☑ Novel artificial textures with anticipated visual properties
- ☑ Numerically efficient
- ☒ Difficult modelling of unseen (abstract) meaningful textures
- ☒ Difficult GPU implementation

Conclusions on BTF Modelling

- Modelling approaches overview

	compres.	enlarg. seamless	A/S separated	spectral synthesis	blockwise impl.	GPU	paral.	unseen data
GMRF 2D	$1:10^5$	Y	Y	–	?	N	Y	Y
GMRF 3D	$1:10^5$	Y	Y	+	?	N	Y	Y
CAR 2D	$1:10^5$	Y	Y	–	Y	Y	Y	Y
CAR 3D	$1:10^5$	Y	Y	+	Y	Y	Y	Y
C-MRF	$1:10^5$	Y	Y	+	Y	Y	Y	Y-
LMP-tiled	$1:10$	Y	Y	+	Y	Y	Y	N
LMPC-til.	$1:10^2$	Y	Y	+	Y	Y	Y	N
G/D/B/M	?	Y	Y	+	Y	N	N	N
hybrid	$1:3$	Y	Y	+	N	N	N	N
tilling	$1:3$	Y	Y	+	Y	Y	N	N
roller	$1:3$	Y	Y	+	Y	Y	N	N

- No ideal method exists. Generally:
  - Better visual quality ⇨ sampling-based methods
  - Higher compression ⇨ probabilistic methods

References

Image Sampling Approaches

- D. Heeger and J. Bergen, *Pyramid Based Texture Analysis/Synthesis*, Proc. ACM SIGGRAPH, pp. 229-238, 1995,
- J. De Bonet, *Multiresolution Sampling Procedure for Analysis and Synthesis of Textured Images*, Proc. ACM SIGGRAPH, pp. 361-368, 1997,
- Y. Xu, B. Guo, and H. Shum, *Chaos Mosaic: Fast and Memory Efficient Texture Synthesis*, Technical Report MSR-TR-2000-32, Microsoft Res., 2000,
- A.A. Efros and W.T. Freeman, *Image Quilting for Texture Synthesis and Transfer*, Proc. ACM SIGGRAPH, pp. 341-346, 2001.
- L. Wei and M. Levoy, *Texture Synthesis over Arbitrary Manifold Surfaces*, Proc. ACM SIGGRAPH, pp. 355-360, 2001,
- M. Ashikhmin, *Synthesizing Natural Textures*, Proc. ACM Symp. Interactive 3D Graphics, pp. 217-226, 2001,
- S. Zelinka and M. Garland, *Interactive Texture Synthesis on Surfaces Using Jump Maps*, Proc. 14th Eurographics Workshop Rendering, pp. 90-96, 2003.

References

Image Sampling Approaches (continue)

- V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, *Graphcut Textures: Image and Video Synthesis Using Graph Cuts*, ACM TOG, vol. 22, no. 3, pp. 277-286, 2003,
- M. Cohen, J. Shade and, S.E.A. Hiller, *Wang Tiles for Image and Texture Generation*, ACM TOG, vol. 22, no. 3, pp. 287-294, July 2003.

BTF Sampling Approaches

- X. Liu, Y. Yu, and H.Y. Shum, *Synthesizing Bidirectional Texture Functions for Real-World Surfaces*, Proc. ACM SIGGRAPH, pp. 97-106, 2001,
- X. Tong, J. Zhang, L. Liu, B. Wang, X. Guo, and H.-Y. Shum, *Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces*, Proc. ACM SIGGRAPH, pp. 665-672, 2002,
- A. Neubeck, A. Zalesny, and L. Van Gool, *Viewpoint Consistent Texture Synthesis*, Proc. Conf. 3D DPVT, pp. 388-395, Sept. 2004.

References

BTF Sampling Approaches (continue)

- K. Zhou, P. Du, L. Wang, Y. Matsushita, J. Shi, B. Guo, and H.-Y. Shum, *Decorating Surfaces with Bidirectional Texture Functions*, IEEE TVCG, vol. 11, no. 5, pp. 519-528, 2005,
- H. Kawasaki, K.-D. Seo, Y. Ohsawa, and R. Furukawa, *Patch-Based BTF Synthesis for Real-Time Rendering*, Proc. IEEE Int'l Conf. Image Processing, vol. 1, pp. 393-396, Sept. 2005,
- C.-S. Leung, W.-M. Pang, C.-W. Fu, T.-T. Wong, and P.-A. Heng, *Tileable BTF*, IEEE TVCG, vol. 13, no. 5, pp. 935-955, 2007,
- P. Somol and M. Haindl, *Novel Path Search Algorithm for Image Stitching and Advanced Texture Tiling*, VSCG , pp. 155-162, 2005,
- M. Haindl and M. Hatka, *A Roller – Fast Sampling-Based Texture Synthesis Algorithm*, VSCG, pp. 80-83, Jan. 2005,
- M. Haindl and M. Hatka, *BTF Roller*, TEXTURE 05, pp. 89-94, Oct. 2005.

References

Reflectance models

- E.P. Lafortune, S.C. Foo, K.E. Torrance, and D.P. Greenberg, *Non-Linear Approximation of Reflectance Functions*, Computer Graphics, vol. 31, pp. 117-126, 1997.
- J. Filip and M. Haindl, *Non-Linear Reflectance Model for Bidirectional Texture Function Synthesis*, ICPR, vol. 1, pp. 80-83, Aug. 2004.
- J. Filip and M. Haindl, *Efficient Image-Based Bidirectional Texture Function Model*, TEXTURE 05, pp. 7-12, Oct. 2005.

BRDF Texels

- J. Filip and M. Haindl, *BTF Modelling Using BRDF Texels*, Int'l J. Computer Math., vol. 84, no. 9, pp. 1267-1283, Sept. 2007.

Compound MRF

- M. Haindl and V. Havlicek, *A Compound MRF Texture Model*, ICPR, Aug. 2010.

References

Probabilistic BTF Models

- M. Haindl and V. Havlicek, *A Multiresolution Causal Colour Texture Model*, Advances in Pattern Recognition, chap. 1, pp. 114-122, Springer-Verlag, Aug. 2000,
- M. Haindl and J. Filip, *Fast BTF Texture Modelling*, TEXTURE 03, Oct. 2003,
- M. Haindl and J. Filip, *A Fast Probabilistic Bidirectional Texture Function Model*, Lecture Notes in Computer Science, vol. 3212, pp. 298-305, Springer, 2004,
- M. Haindl and J. Filip, *Extreme Compression and Modeling of Bidirectional Texture Function*, IEEE TPAMI, vol. 29, no. 10, pp. 1859-1865, Oct. 2007,
- M. Haindl, J. Filip, and M. Arnold, *BTF Image Space Utmost Compression and Modelling Method*, ICPR, vol. 3, pp. 194- 197, Aug. 2004.

Hybrid Models

- M. Haindl, J. Grim, P. Pudil, and M. Kudo, *A Hybrid BTF Model Based on Gaussian Mixtures*, ICPR , pp. 95-100, Oct. 2005.

References

Mixture Models

- J. Grim and M. Haindl, *Texture Modelling by Discrete Distribution Mixtures*, Computational Statistics Data Analysis, vol. 41, nos. 3/4, pp. 603-615, Jan. 2003.

BTF Editing

- J. Kautz, S. Boulos, and F. Durand, *Interactive Editing and Modelling of Bidirectional Texture Functions*, ACM Trans. Graphics, vol. 26, no. 3, p. 53, 2007,
- N. Menzel and M. Guthe, *g-BRDFs: An Intuitive and Editable BTF Representation*, Computer Graphics Forum, vol. 28, no. 8, pp. 2189 - 2200, 2009,
- M.Haindl and V. Havlíček, *Texture Editing Using Frequency Swap Strategy*, CAIP 2009.
- M.Haindl and M. Hatka, *Near-Regular Texture Synthesis*, CAIP 2009.

Appendix
2D CAR Model

$$p(Y\,|\,\gamma,\sigma^{-2}) = (2\pi\sigma^2)^{-\frac{(MN-1)}{2}} \exp\left\{\frac{-1}{2} tr\left\{\sigma^{-2}\begin{pmatrix}-\alpha\\ \gamma^T\end{pmatrix}^T \tilde{V}_{MN-1}\begin{pmatrix}-\alpha\\ \gamma^T\end{pmatrix}\right\}\right\}$$

α a unit vector

$$\tilde{V}_{r-1} = \begin{pmatrix}\tilde{V}_{y(r-1)} & \tilde{V}^T_{xy(r-1)}\\ \tilde{V}_{xy(r-1)} & \tilde{V}_{x(r-1)}\end{pmatrix}$$

$$\tilde{V}_{a(r-1)} = \sum_{k=1}^{r-1} A_k\, A_k^T$$

$$\tilde{V}_{ab(r-1)} = \sum_{k=1}^{r-1} A_k\, B_k^T$$

Appendix
2D CAR Model – Parameters Estimation

$$Y_r = \gamma X_r + e_r$$

$$\gamma = [a_1,\ldots,a_\eta] \qquad \eta = card(I_r^c) \qquad \sigma^2$$

$$\begin{aligned}
\hat{\gamma}^T_{r-1} &= V^{-1}_{x(r-1)} V_{xy(r-1)}\\
\hat{\sigma}^2_{r-1} &= \frac{\lambda_{(r-1)}}{\beta(r)}\\
V_{z(r-1)} &= \tilde{V}_{z(r-1)} + V_{z(0)}\\
\lambda_{(r-1)} &= V_{y(r-1)} - V^T_{xy(r-1)} V^{-1}_{x(r-1)} V_{xy(r-1)}
\end{aligned}$$

Appendix
2D CAR Model – Optimal Model

M_i – model

$$\max_j\{p(M_j|Y^{(r-1)})\}$$

$$\begin{aligned}
D_{j(r-1)} &= -\frac{d}{2}\ln|V_{x(r-1)}|\\
&- \frac{\beta(r)-d\eta+d+1}{2}\ln|\lambda_{(r-1)}|\\
&+ \frac{d^2\eta}{2}\ln\pi\sum_{i=1}^{d}\left[\ln\Gamma\left(\frac{\beta(r)-d\eta+d+2-i}{2}\right)\right.\\
&- \left.\ln\Gamma\left(\frac{\beta(0)-d\eta+d+2-i}{2}\right)\right]
\end{aligned}$$

Appendix
Gaussian Mixture Model - Analysis

$$p(Y_{\{r\}}) = \sum_{i=1}^{K} p_i\, p(Y_{\{r\}}\,|\,\mu_i,\Sigma_i)$$

$$p(Y_{\{r\}}\,|\,\mu_i,\Sigma_i) = \frac{|\Sigma_i|^{-\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\exp\left\{-\frac{(Y_{\{r\}}-\mu_i)^T\Sigma_i^{-1}(Y_{\{r\}}-\mu_i)}{2}\right\}$$

$$p(Y_{\{r\}}\,|\,\mu_i,\Sigma_i) = \prod_{\forall s\in\{r\}} p(Y_s\,|\,\mu_{i,s},\sigma_{i,s})$$

$$L = \frac{1}{|I|}\sum_{Y_{\{r\}}:\forall r\in I}\ln\sum_{i=1}^{K} p_i\, p(Y_{\{r\}}\,|\,\mu_i,\Sigma_i)$$

Appendix
Gaussian Mixture Model - Synthesis

$$
\begin{aligned}
p(Y_s|Y_C) &= \sum_{i=1}^{K} W_i(Y_C)\, p(Y_s \,|\, \mu_{i,s}, \sigma_{i,s}) \\
W_i(Y_C) &= \frac{w_i\, p(Y_C|\mu_i, \sigma_i)}{\sum_{j=1}^{K} w_j\, p(Y_C \,|\, \mu_j, \sigma_j)} \\
p(Y_C|\mu_i, \sigma_i) &= \prod_{s \in C} p(Y_s \,|\, \mu_{i,s}, \sigma_{i,s})
\end{aligned}
$$

$C \subset \{r\}$ $\qquad$ $s \in \{r\} - C$