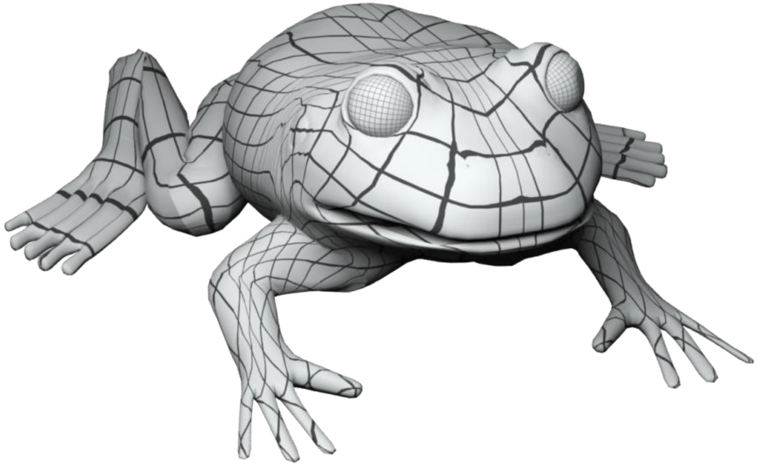
CHEMNITZ UNIVERSITY OF TECHNOLOGY

Efficient encoding of texture coordinates guided by mesh geometry



Libor Váša
Guido Brunnett

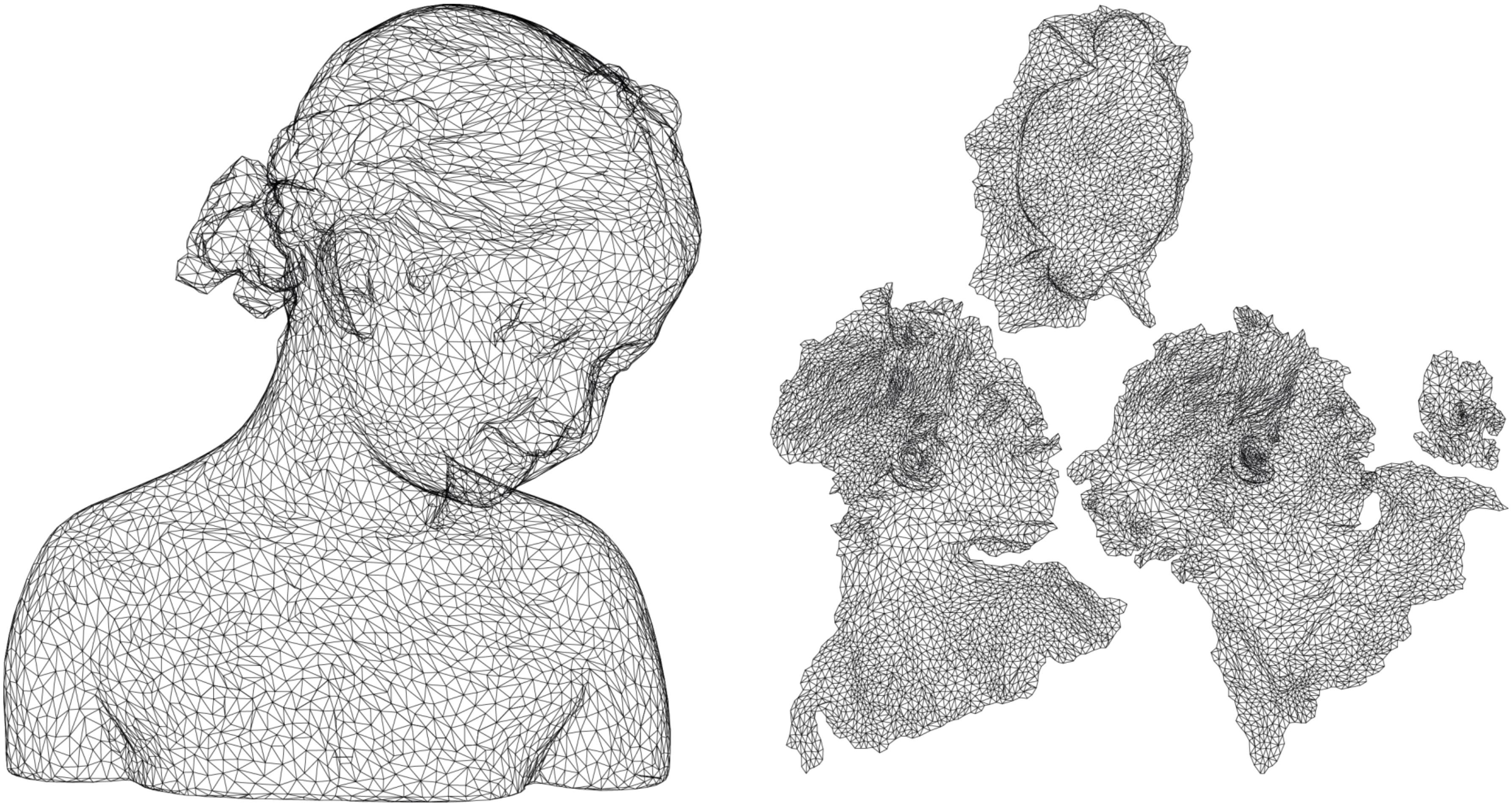
Symposium on Geometry Processing 2014, Cardiff

Task: Texture coordinates compression

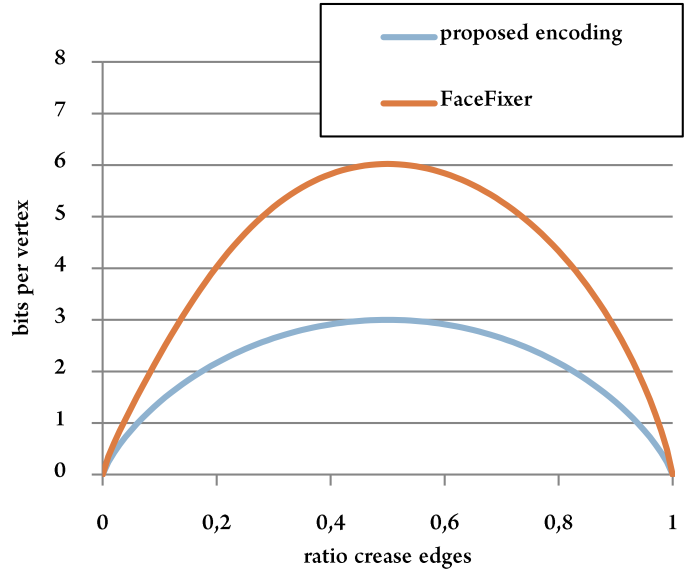
- Texturing = parameterization of the surface
  - UV coordinates associated with each corner
  - Usually the same for all corners of a vertex (except for crease edges)

- Mesh connectivity + geometry already transmitted
- Encoding of texture connectivity
- Lossy encoding of texture geometry

Mesh parameterization



Texture connectivity



- Closely related to mesh connectivity
- Mostly derived by a series of cuts
- => encoding: 1 bit per edge (cut or not)
- Better theoretical performance than other approaches, such as vertex bits etc.
- Lacks support for some unlikely/unwanted phenomena, such as
  - Welding of triangles that were not connected in mesh
  - Non-manifold texture connectivity

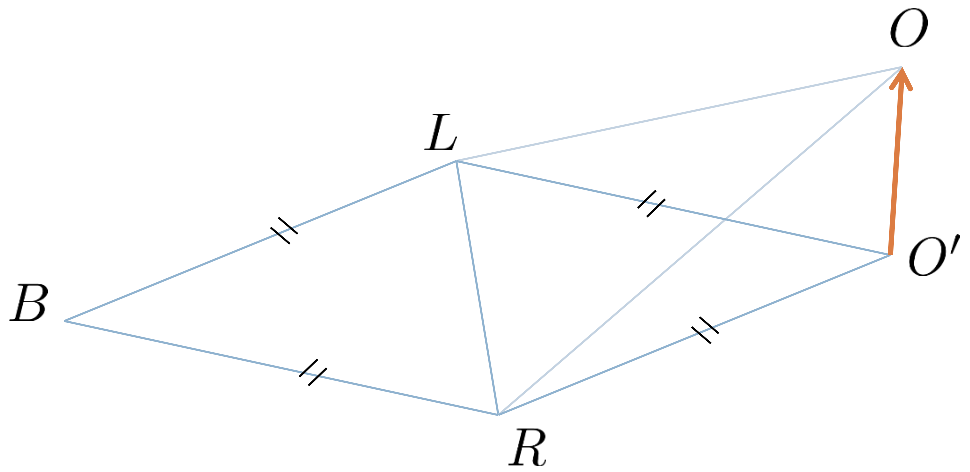
Data compression: prediction

"What can be guessed does not have to be transmitted"

input:	123456789		
	~30 bits		
Prediction:	123314159	Residual	142630
			~20 bits
Better prediction:	123456314	Residual	475
			~10 bits

Compression of mesh geometry

- Paralelogram predictor



$$O' = L + R - B$$

[Touma & Gotsman, 1998]

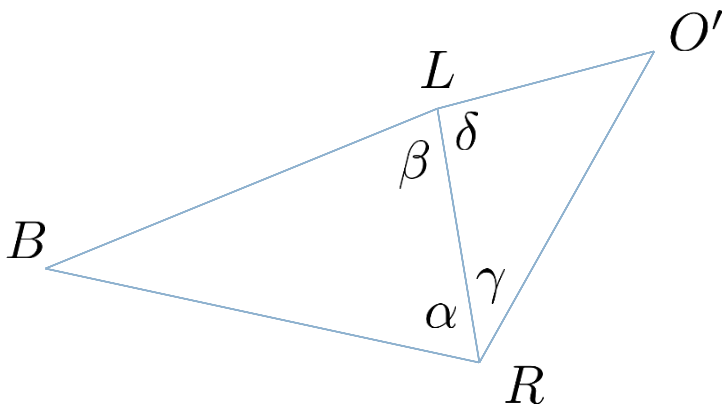
Weighted parallelogram predictor

- Need more freedom

$$O' = w_1 L + w_2 R + (1 - w_1 - w_2) B$$

$$O' - B = w_1 (L - B) + w_2 (R - B)$$

$$w_1 = \frac{\cot \alpha + \cot \gamma}{\cot \delta + \cot \gamma} \qquad w_2 = \frac{\cot \delta + \cot \beta}{\cot \delta + \cot \gamma}$$



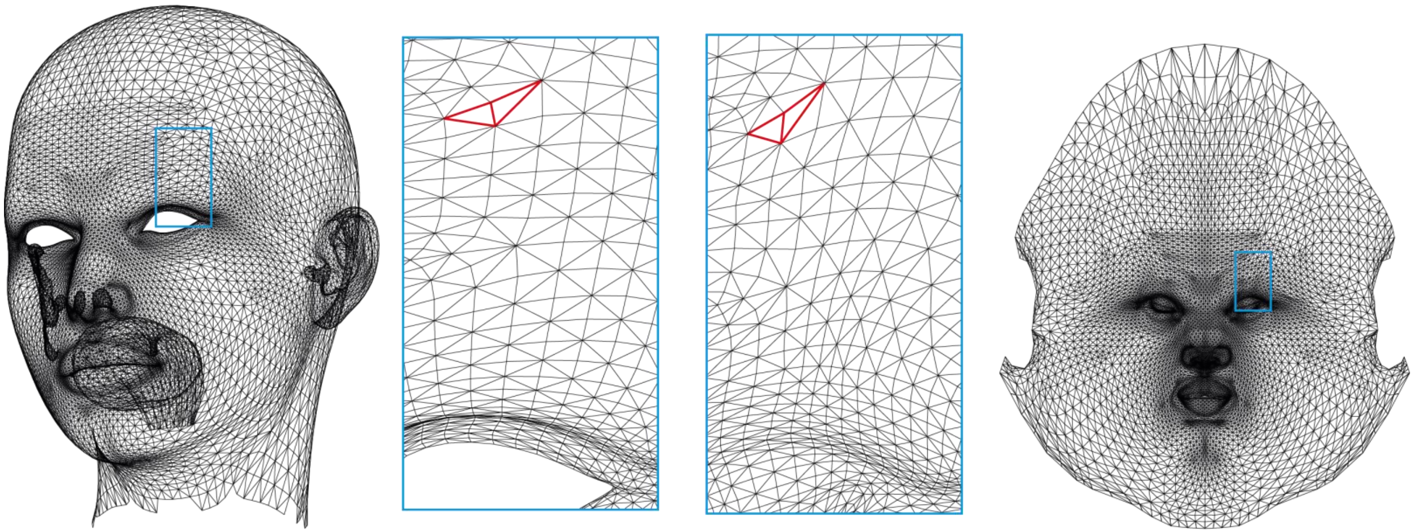
$$\delta' = \beta' = \frac{2\pi}{d_L}$$

$$\alpha' = \gamma' = \frac{2\pi}{d_R}$$

[Váša & Brunnett 2013]

WPP for texture coordinates

- Mesh geometry already transmitted
- Each texture triangle has a corresponding mesh triangle
- Assumption – parameterization is at least partially conformal

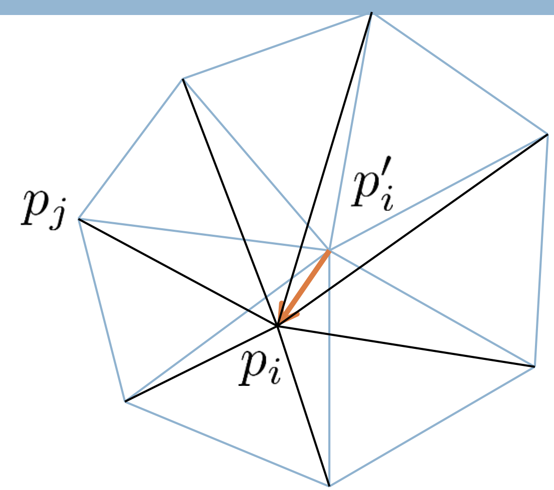


=> use angles from mesh geometry as prediction of angles in texture geometry

Laplacian coding of mesh geometry

- Predictor for mesh encoding

$$p'_i = \frac{1}{|N(i)|} \sum_{j \in N(i)} p_j$$



- Residual:

$$r_i = p'_i - p_i = \frac{1}{|N(i)|} \sum_{j \in N(i)} (p_j - p_i)$$

- Encoding + Decoding

recosntruction

$$\mathbf{r} = L^* \mathbf{p} \qquad \hat{\mathbf{p}} = L^{*-1} \hat{\mathbf{r}}$$

residuals

Combinatorial Laplacian

[Sorkine et. al., 2003]

Laplacian coding of texture coordinates

$$\mathbf{r} = L^* \mathbf{p} \qquad \hat{\mathbf{p}} = L^{*-1} \hat{\mathbf{r}}$$

- Use a geometric Laplacian instead of combinatorial
  - Mean value Laplacian, cotan Laplacian …
  - Weights make the prediction more accurate
- Angles/edge lengths from mesh
  - Normalize angles (inner vertex sum = 2π, border sum=π)
- Structure (neighbourhoods) from texture connectivity
  - Crease edges

Results

- Reduced entropy of residuals
  - Lower data rate at the same distortion
  - Lower distortion at the same data rate

Proposed

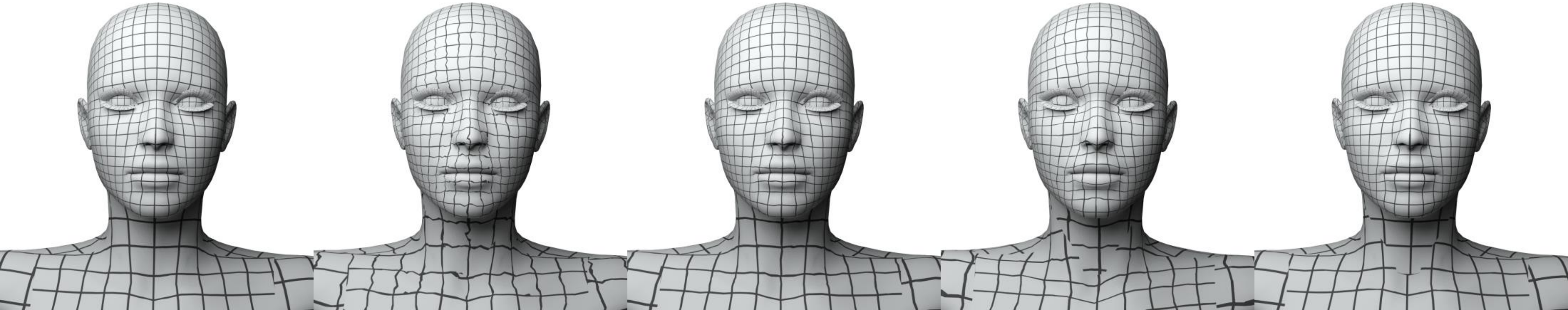
original

Parallelogram prediction

Weighted parallelogram prediction

Combinatorial Laplacian

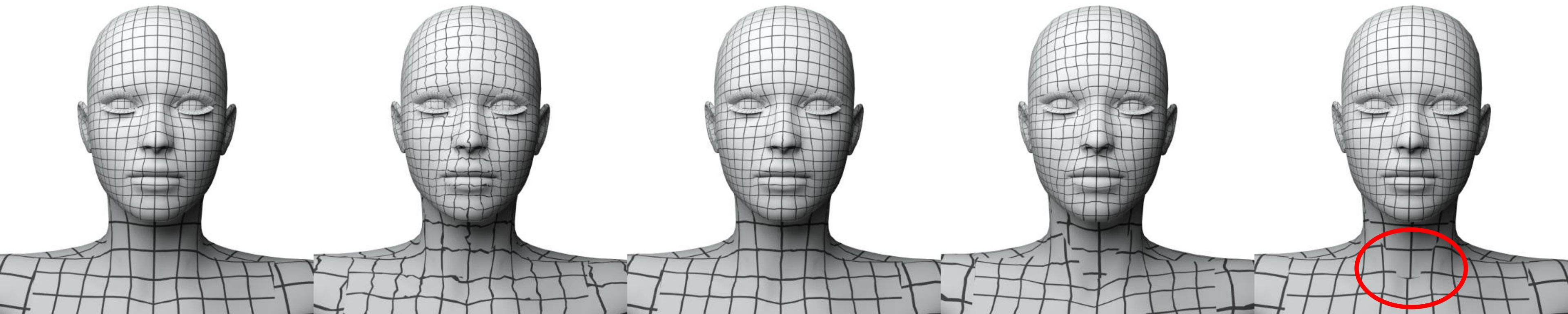
Geometric Laplacian



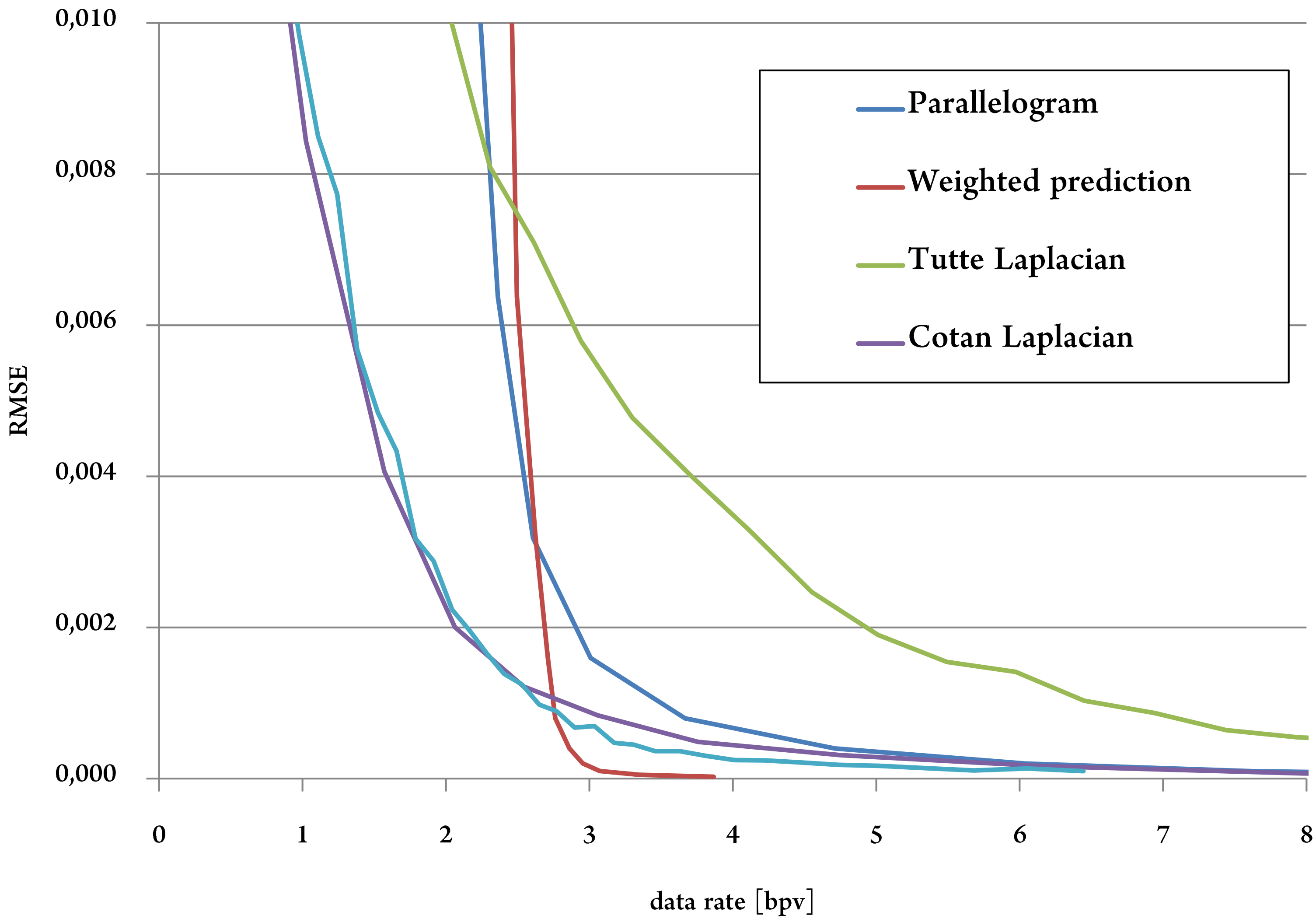
Results

- Laplacian encoding
  - Slower (solving a sparse linear system at decoder)
  - More efficient for low bitrates
  - Distortion visible on crease edges

- Weighted parallelogram
  - Small slowdown with respect to pure paralellogram
  - Distortion more uniformly distributed



Typical result – DAZ dataset



Various parameterizations

model	parameterization	RMSE	Parallelogram	Weighted prediction	Tutte	Cotan	MV
Horse	ABF	0,0015	2,77	2,59	8,47	2,00	2,12
		0,0001	6,33	2,95	16,57	4,87	6,30
	ABF++	0,0015	2,80	2,53	9,05	2,01	2,13
		0,0001	6,37	2,99	16,63	4,83	6,25
	DPBF	0,0015	2,77	2,55	8,05	2,87	2,63
		0,0001	6,45	3,23	16,70	9,66	10,13
	LSCM	0,0015	2,86	2,63	9,30	2,69	2,63
		0,0001	6,48	2,99	18,04	6,02	7,33
	HLSCM	0,0015	2,82	2,61	9,31	2,70	2,63
		0,0001	6,48	2,99	18,05	5,42	7,32
Victoria	manual	0,0008	7,47	4,83	12,48	9,63	9,58
		0,0001	13,74	9,49	19,84	16,28	15,73

Data rates [bpv]

Conclusion

- Specialized algorithms proposed for texture coordinates compression
- Mesh geometry can be efficiently exploited for more efficient compression of texture geometry
- Parallelogram prediction and Laplacian based coding can be extended

Thank you

http://www.tu-chemnitz.de/informatik/GDV/

Efficient encoding of texture coordinates guided by mesh geometry

Libor Váša
Guido Brunnett

CHEMNITZ UNIVERSITY OF TECHNOLOGY