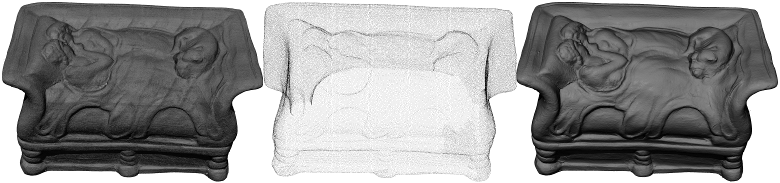
Self-similarity for accurate compression of point sampled surfaces

Julie Digne[1] Raphaëlle Chaine[1] Sébastien Valette[2]

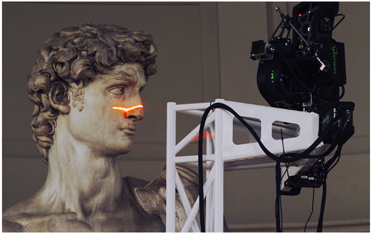
[1] LIRIS - CNRS - Université Lyon 1; [2] CREATIS - INSA de Lyon

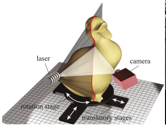
December 19th, 2013

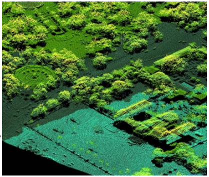


Context

- Increasing quantity of points
- Increasing measurement accuracy
- Increasing ways of acquiring surfaces (laser, photogrammetry...)
- Low cost acquistion devices (kinect...)







Why store points?

- A mesh is *one* way to represent a surface
- The points correspond to the raw information without any interpretation
- Some applications do not need connectivity
- Difficulty with unorganized points: no sampling theorem, no structure!

How do we store pointsets without destroying their input accuracy?

Idea: modify the point cloud *below the surface accuracy*

- The surface comes with an input accuracy, the acquisition devices is accurate to a given precision.
- Idea: Resample the surface with local patterns
- Compress these local patterns

Two errors: resampling error and compression error. We can control them!

Outline

Previous Work

Compression

Decompression

Results

Conclusion and Perspectives

Outline

Previous Work

Compression

Decompression

Results

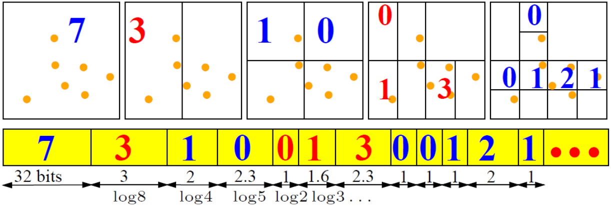
Conclusion and Perspectives

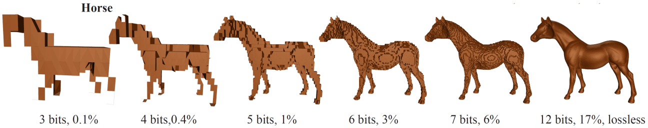
Compression for meshes

- Mesh compression dates back to Deering 1995
- Single resolution meshes compressed by valence coding (Touma, Gotsman, 1998)
- Progressive compression (Alliez and Desbrun 2001)
- Compression via Wavelets of a shape remeshed with subdivision connectivity [Guskov et al. 2000], [Peyré and Mallat 2005].

Point Cloud Compression

- Coordinate quantification via octree coding (Gandoin and Devillers 2002, Schnabel and Klein 2006, Smith et al. 2012)





Images from Gandoin and Devillers 2002

Point Cloud Compression for Rendering

Primitive based compression Schnabel, Möser, Klein, 2007-2008

- Shape segmentation based on primitive regression (RANSAC)
- Height fields over the primitives are computed
- These height-fields are decomposed via vector quantization

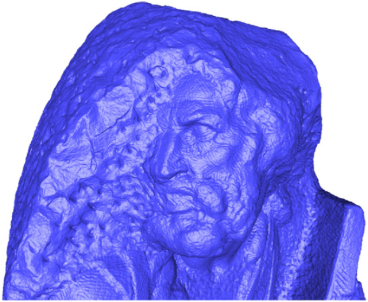
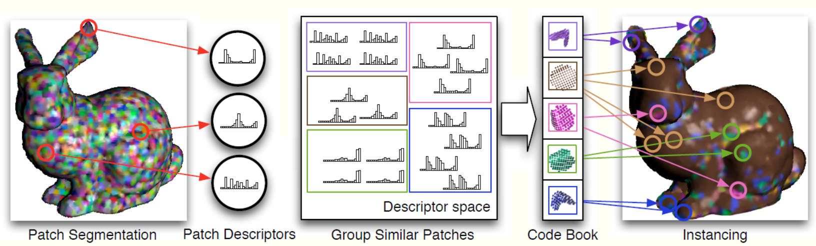


Image from Schnabel et al. 2008, 3.31 bits per point

Attempt at using self-similarity for compression

Hubo et al. 2008

- Select a subset of points
- Define patches, group them by similarity
- Replace each patch by its codebook



Self-similarity for surfaces

- Similarity for the analysis of the structures in a surface
- Symmetry or repeated structures in the surface (Mitra et al 2006, Pauly et al. 2012)
- Denoising for meshes (Yoshizawa 2006) and point clouds (Digne 2012)

Similarity based denoising of point clouds

Patch-based approaches for signal and image processing

- Texture synthesis (Efros 99), Non local means (Buadès et al. 2005).

Patch-based approaches for signal and image processing

- Texture synthesis (Efros 99), Non local means (Buadès et al. 2005).
- Compressive sensing theory (Candès et al. 2006): there exists spaces, in which the signals would be sparsely represented, that are especially well suited for processing the signals.

Patch-based approaches for signal and image processing

- Texture synthesis (Efros 99), Non local means (Buadès et al. 2005).
- Compressive sensing theory (Candès et al. 2006): there exists spaces, in which the signals would be sparsely represented, that are especially well suited for processing the signals.
- Sparse regularization for image analysis, inpainting... [Elad et al. 2006] [Mairal 2009] The K-SVD algorithm

Patch-based approaches for signal and image processing

- Texture synthesis (Efros 99), Non local means (Buadès et al. 2005).
- Compressive sensing theory (Candès et al. 2006): there exists spaces, in which the signals would be sparsely represented, that are especially well suited for processing the signals.
- Sparse regularization for image analysis, inpainting... [Elad et al. 2006] [Mairal 2009] The K-SVD algorithm
- Can we adopt a similar approach for surfaces?

Outline

Previous Work

Compression

Decompression

Results

Conclusion and Perspectives

Compression Summary

- Pick a subset of points (the *seeds*)
- Compute a local parameterization of the seeds
- Describe the surface in these local neighborhoods as a local patch
- Compute a dictionary and a sparse decomposition of these local patches on this dictionary
- Encode the seeds, their local parameterizations, the dictionary and the coefficients.

Working assumptions

- *Topological condition:* R must be set such that $\mathcal{P}$ can be covered by the set of R-neighborhoods corresponding to a subset of seed points in $\mathcal{P}$. Additionally, each R-neighborhood should delimit a topological disk on the underlying surface $\mathcal{M}$.
- *Sampling condition:* The R-neighborhood of a seed point must contain enough points so that a meaningful patch of surface can be computed.
- *Noise level:* The noise magnitude is strictly below radius R.

NB: The points may (or may not) be equipped with a normal.

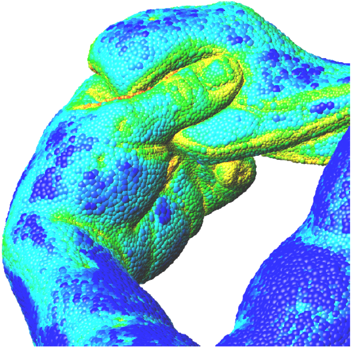
Seed selection

- We select a subset of the points, the *seeds* that will serve as anchors to define local patches
- The subset of the seeds $\mathcal{S}$ satisfies:

$$\forall p \in \mathcal{P}, \exists s \in \mathcal{S}, \|p - s\| \leq R.$$

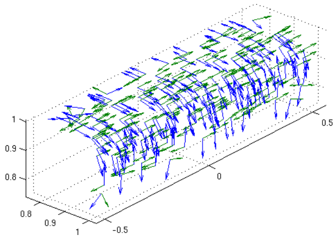
- What is a good covering of the points?
- Outliers: a patch will be created for each outlier $\Rightarrow$ minimum coverage threshold to avoid it.

- Seeds selected in a dart-throwing fashion.



Choice of a local coordinate system

- p a point of the surface, let (p_i) its neighbors
- $\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i$; $C_I = \frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{p})^T \cdot (p_i - \bar{p})$.
- Unoriented case: n_i is the eigenvector corresponding to the least eigenvalue of the local covariance matrix
- $\vec{n}_m = \frac{1}{N}\sum_{i=1}^{N} \vec{n}_i$; $C_{II} = \frac{1}{N}\sum_{i=1}^{N}(\vec{n}_i - \vec{n}_m)^T \cdot (\vec{n}_i - \vec{n}_m)$.
- Eigenvectors of CII give the local parameterization in the tangent plane
- If the normal is oriented, one picks the local frame orientation accordingly.

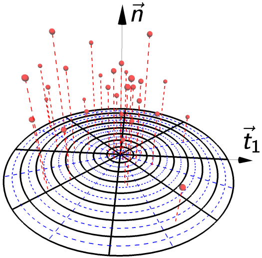


Neighborhood description

- Each point p is equipped with a normal $\vec{n}$ and a tangent vector $\vec{t}_1$
- a radial grid $(r_i, \theta_j)_{i,j=0\cdots N_{bins}-1}$ such that:

$$r_i = (\frac{1}{2}+i)\cdot\frac{R}{N_{bins}}; \theta_j = j\cdot\frac{2\pi}{N_{bins}}$$

- Interpolate linearly the local height field on this grid.



This simple neighborhood description already implies an error: we must ensure that this error is below the input scanner accuracy.

Self-similarity compression

- All local patches are represented in a comparable way
- A dictionary is built upon which all patches will be represented
- The compression will consist in a set of parameterized seeds, a small dictionary and the (sparse) coefficients of the patch decomposition
- The dictionary is found by the K-SVD algorithm

A short summary of K-SVD

- The K-SVD algorithm is a method for building representations of finite discretized signals as sparse linear combinations over an ad hoc dictionary (Mairal, Sapiro, Elad...)
- Y is a $k \times n$ matrix, whose columns are n training signals $(y_i)_{i=1,\cdots n}$
- Goal: Find a dictionary D, composed of d signal atoms, over which each signal y_i can be represented as a linear combination of the dictionary atoms d_i.
- Both X and D are solved for by computing:

$$\min_{D,X} \|Y - DX\| \text{ s.t. } \forall i, \|x_i\|_0 \leq T_0$$

- An iterative approach that alternates between two steps
  - ***Sparse coding*** of the examples based on the current dictionary
  - ***Update of the dictionary*** so as to better fit the data

Sparse Coding stage

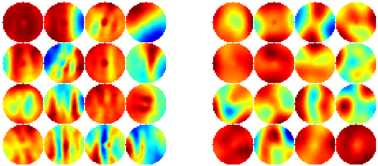
- $\mathbf{D}$ is fixed, compute the best representation x_i of sample y_i
- Find x_i minimizing $\|y_i - \mathbf{D}x_i\|_2^2$ s.t. $\|x_i\|_0 \leq T_0$
- Can be done using a pursuit algorithm (e.g. Orthogonal Matching Pursuit)

Dictionary Update stage

- The update will be done atom by atom.
- $\|\mathbf{Y}-\mathbf{DX}\|_F^2 = \|\mathbf{Y}-\sum_{j=1}^{N} d_j x_T^j\|_F^2 = \|\mathbf{Y}-\sum_{j=1,j\neq k}^{N} d_j x_T^j - d_k x_T^k\|_F^2$
- $E_k = \mathbf{Y}-\sum_{j=1,j\neq k}^{N} d_j x_T^j$ error obtained by omitting atom d_k in the decomposition
- Finally solve for :

$$\|E_k - d_k x_T^k\|_F \text{ w.r.t. } d_k, x_T^k$$

- Solve using SVD? if so sparsity not enforced -> build an auxiliary matrix to enforce sparsity



Dictionaries built for two different shapes: a geometrical one (the mire, left) and a fine art one (the Lovers, right).

The atoms are shown by order of importance (total absolute weight in the linear decompositions).

Further compression

- The coefficients are compressed via a scalar quantization followed by entropy coding.
- The seeds are compressed via an octree-based compression (Gandoin and Devillers, 2002).
- The local parameterization is represented as three Euler angles: we encode the difference of the true parametrization with the parameterization estimated on the seeds.
- The dictionary is losslessly compressed.

Compression summary

This Compression scheme yields 4 kinds of data :

- the seeds (three coordinates)
- three Euler angles difference giving the local parameterization
- the dictionary (a $N_{atoms} \times N_{bins}^2$ matrix)
- the decomposition coefficients

As an example, For the Lovers of Bordeaux, bitstream sizes are 312KB for the seeds coordinates and 297KB for the angle differences, both quantized on 8 bits. The dictionary and coefficients are respectively encoded on 18KB and 507KB.

Outline

Previous Work

Compression

Decompression

Results

Conclusion and Perspectives

First step: decompressing the seeds and the patches

1. Decompress the seed positions
2. Decompress the Euler angles and the dictionary
3. Decompress the coefficients X
4. Reconstruct the patches: $P_{rec} = D * X$

Second step: recover the surface from the patches

- Each bin (r, θ) of the sampling pattern $\mathcal{F}_s(r, \theta)$ yields potentially a point:
$$(r\cos(\theta), r\sin(\theta), \mathcal{F}_s(r, \theta))$$
- Translation in the global coordinate system as: $s + r\cos(\theta)t_1(s) + r\sin(\theta)t_2(s) + \mathcal{F}_s(r, \theta)n(s)$. Problem: each point can be covered several times!
- Consolidate the reconstructed point cloud in the overlapping areas.

Outline

Previous Work

Compression

Decompression

Results

Conclusion and Perspectives

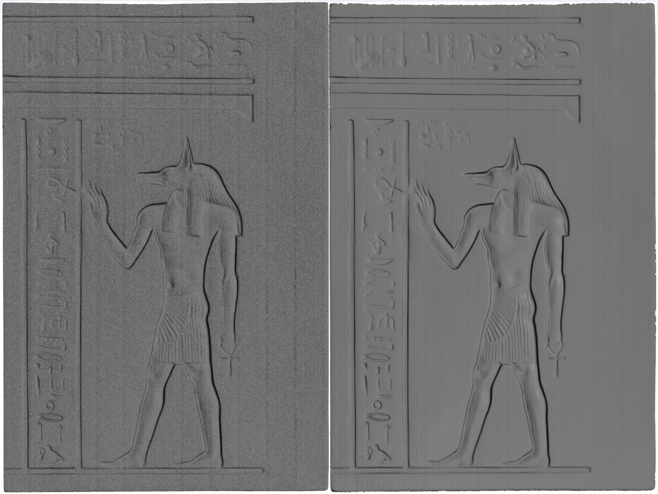


Figure : Compression and decompression of the Anubis point set (Left:original, right: decompression)

Figure : Rendering of the original (left) and decompressed Stanford St Matthew (right), both have 93.4 million points

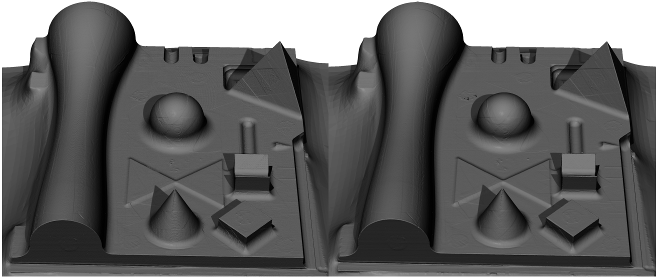


Figure : Original mire pointset (top) and decompression (bottom). Both pointsets were reconstructed using the Screened Poisson Reconstruction [Kazdhan, 2013]

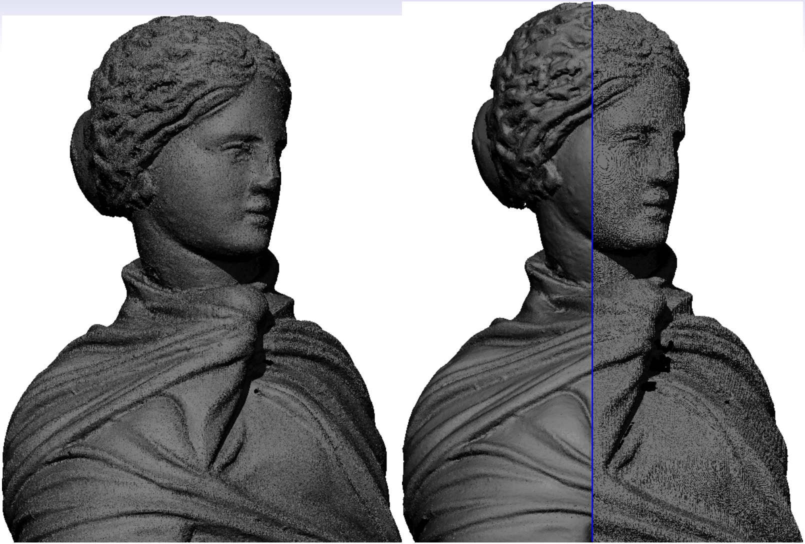
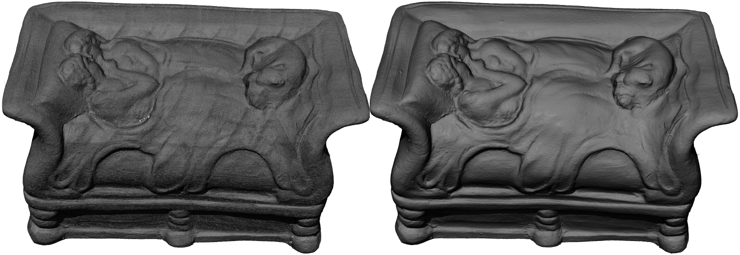


Figure : Comparison with octree coding. Left : original point cloud. Right : comparative decompression. Even with more bits per point (4.83 against 0.6 in our method), the right part encoded with Gandoin et al. is less accurate than our approach (left part).



Breaking the working assumptions

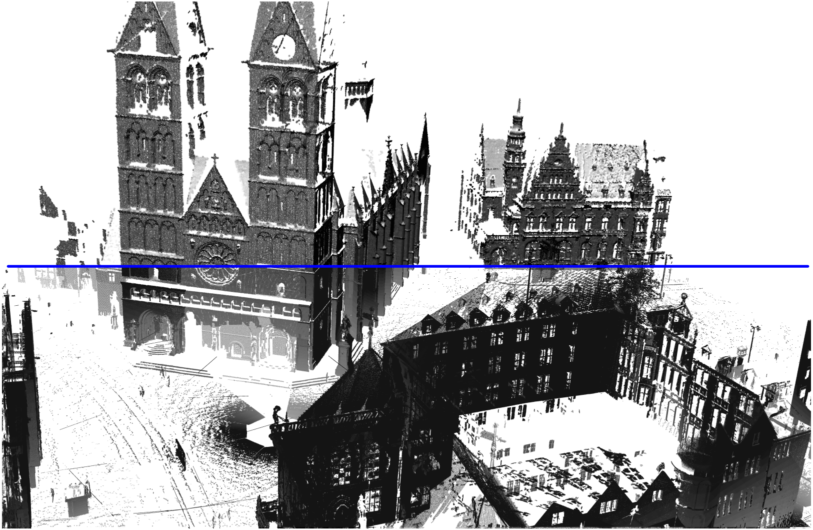


Figure : The Bremen point cloud (bottom) and the decompressed result (top)

Pointset	number of points	R	compressed size (bytes)	RMSE (% of diagonal)	Percentage of points with error above sampling precision	bpp
Anubis	$9,9M$	$0.7mm$	$1,201,636$	0.01mm (0.003%)	1.23%	0.96
Lovers of Bordeaux	$15,8M$	$0.5mm$	$1,152,245$	0.01mm (0.006%)	0.86%	0.59
Mire	$16,1M$	$0.6mm$	$1,480,118$	0.03mm (0.011%)	1.30%	0.73
Tanagra	$16,4M$	$0.7mm$	$1,238,271$	0.01mm (0.004%)	1.56%	0.60
David	$28,2M$	$10mm$	$2,150,711$	0.24mm (0.004%)	0.75%	0.61
Bremen	$69,9M$	$18cm$	$6,699,915$	1.48cm (0.005%)	not available	0.76
St Matthew	$93,5M$	$3mm$	$9,780,886$	0.05cm (0.002%)	not available	0.83

Figure : Compression performance

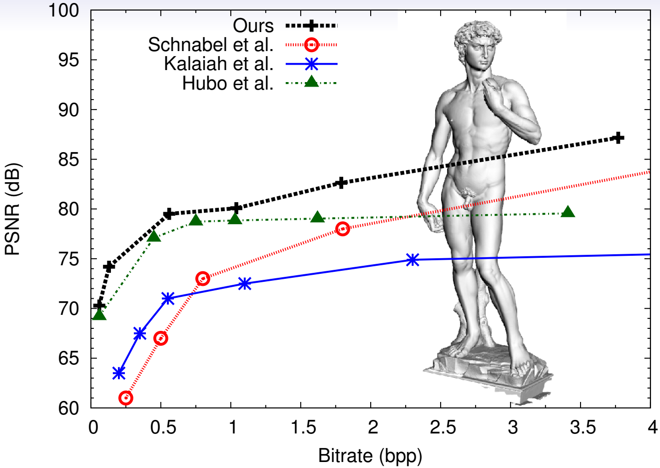


Figure : Comparison with previous works in terms of rate/distortion on the David model. The different bitrates were obtained by increasing the radii of the patches and the size of the descriptors.

Limitations

- It is not a lossless compression scheme (not designed as one)
- Huge computation times still (no parallelization)
- The largest part of the error is caused by the resampling pattern
- Outliers, noise, holes

Outline

Previous Work

Compression

Decompression

Results

Conclusion and Perspectives

Conclusion and Perspectives

- A compression pipeline that targets the precision and preservation of details
- Still a lot of possible improvements:
  - Find a better (cleverer?) way to cover the surface.
  - The resampling pattern could be improved.
  - Better handling of sharp edges.

Thanks to Pierre-Marie Gandoin for providing his octree-based compression code.

More details in *Self-similarity for accurate compression of point sampled surfaces*, J. Digne, R. Chaine, S. Valette, to appear in Computer Graphics Forum, Proceedings Eurographics 2014.