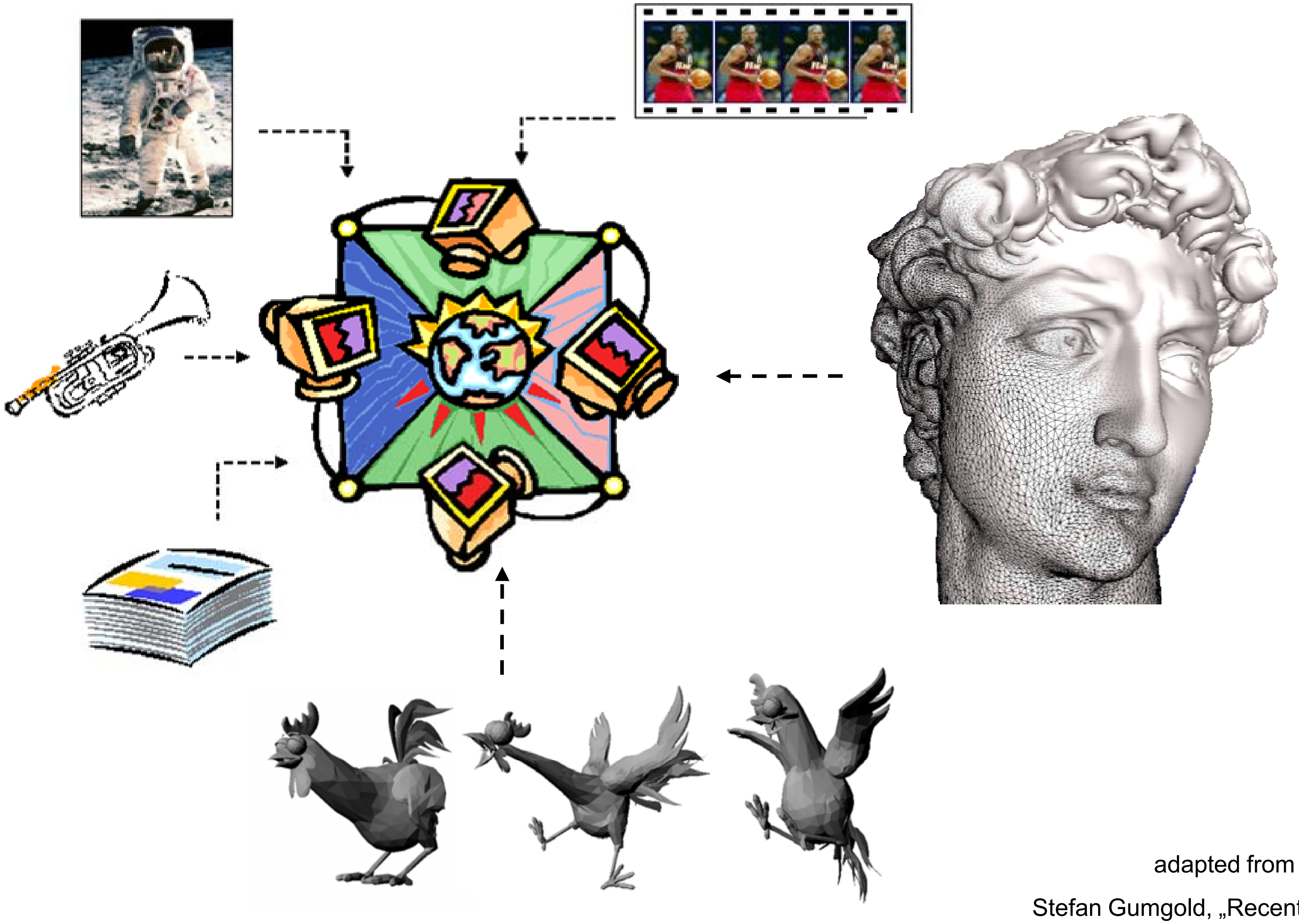
Compression of 3D Meshes: Application, Approaches, Standards

N. Stefanoski[1], L. Vasa[2], J. Ostermann[1]

[1] Leibniz Universität Hannover,
Institut für Informationsverarbeitung (TNT)

[2] University of West Bohemia, Pilsen
Department of Computer Science and Engineering



adapted from

Stefan Gumgold, „Recent Advances in Compression of 3D Meshes“

Outline

- Introduction
- Mesh Distortion Measures
- Compression of Static Meshes
- Compression of Dynamic Meshes
- Summary and Conclusion

Motivation

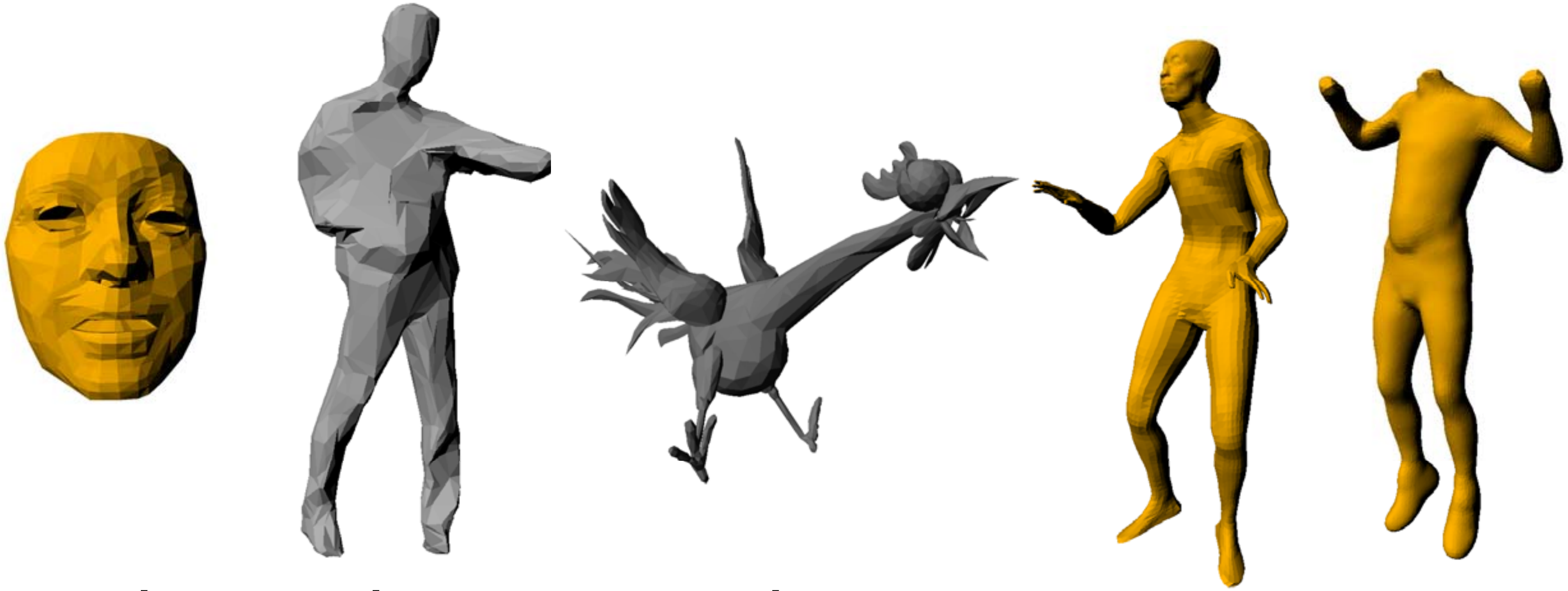
- Visualization of static and dynamic (time varying) geometric data in networked environments




# vertices:	**~0.7k**	**~1.4k**	**~435k**	**~500k**
# raw size: (binary)	**8 kB**	**17 kB**	**8 MB**	**9.5 MB**

Motivation

- Visualization of static and dynamic (time varying) geometric data in networked environments



# vertices:	~0.5k	~1k	~3k	~7k	~15k
# raw rate: (binary)	0.5 MBit/s	1 MBit/s	2.8 MBit/s	6.5 MBit/s	15 MBit/s

General requirements

- fast download and efficient storage → efficient compression
- short latency and robust bit streams → scalable compression

➧ Demand for efficient and scalable compression techniques

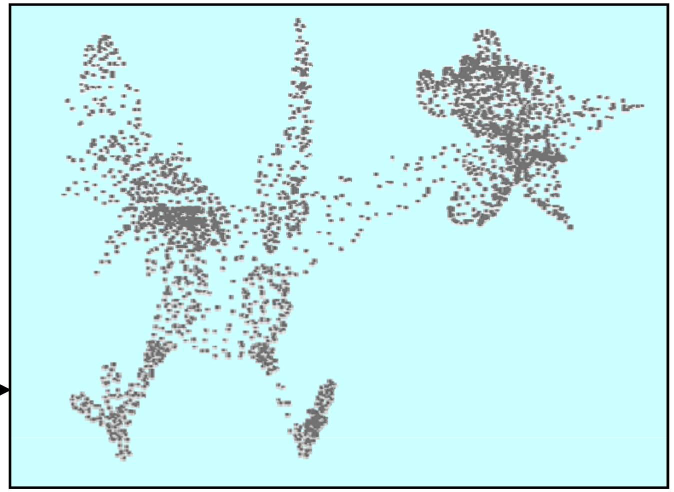
Applications

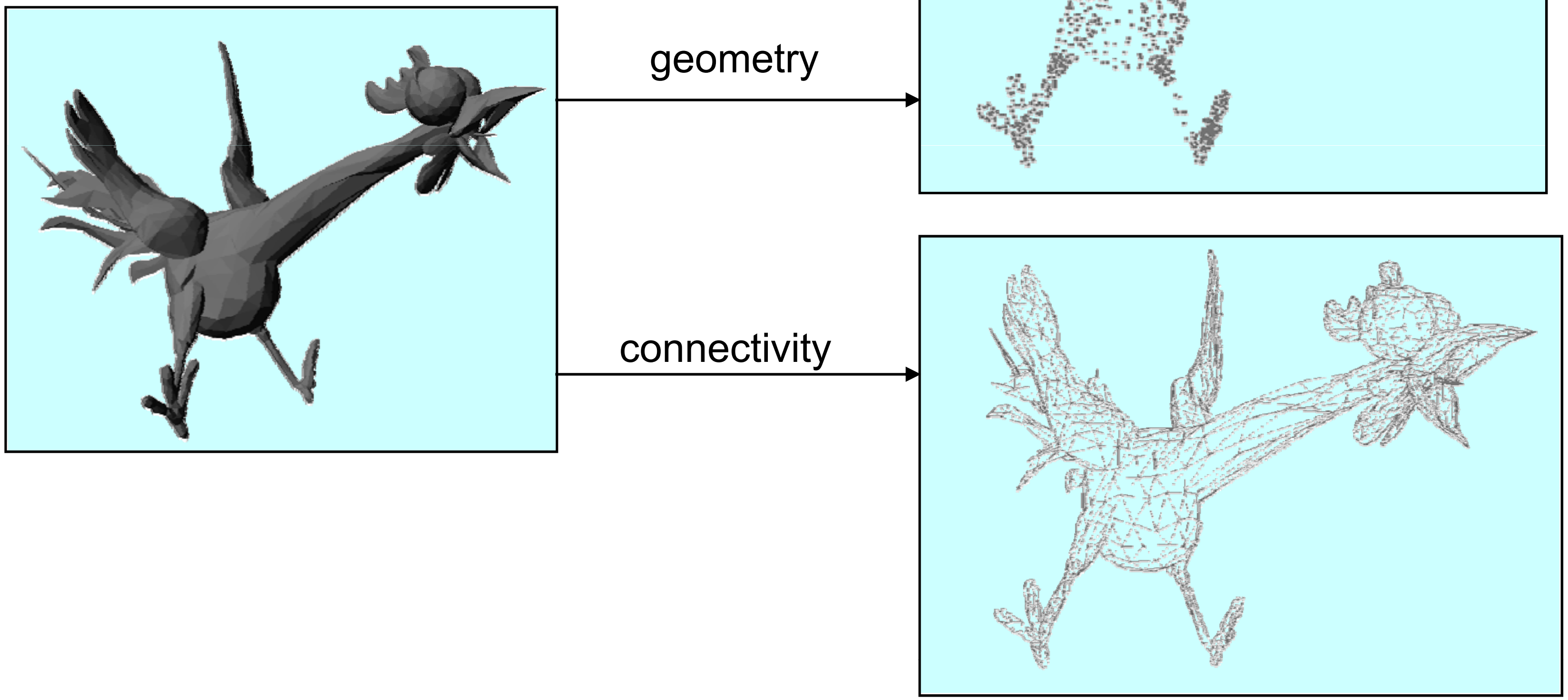
- Virtual Museums
- Cultural Heritage
- Medicine
- CAD/GIS
- Video Games
- 3D Animation Movies
- 3D Cinema/Television
- …

Introduction

Static Mesh

- geometry: 3D vertex points
- connectivity: connections between vertices (def. faces)
- + face and vertex attributes like colors, texture coordinates, normals, etc.





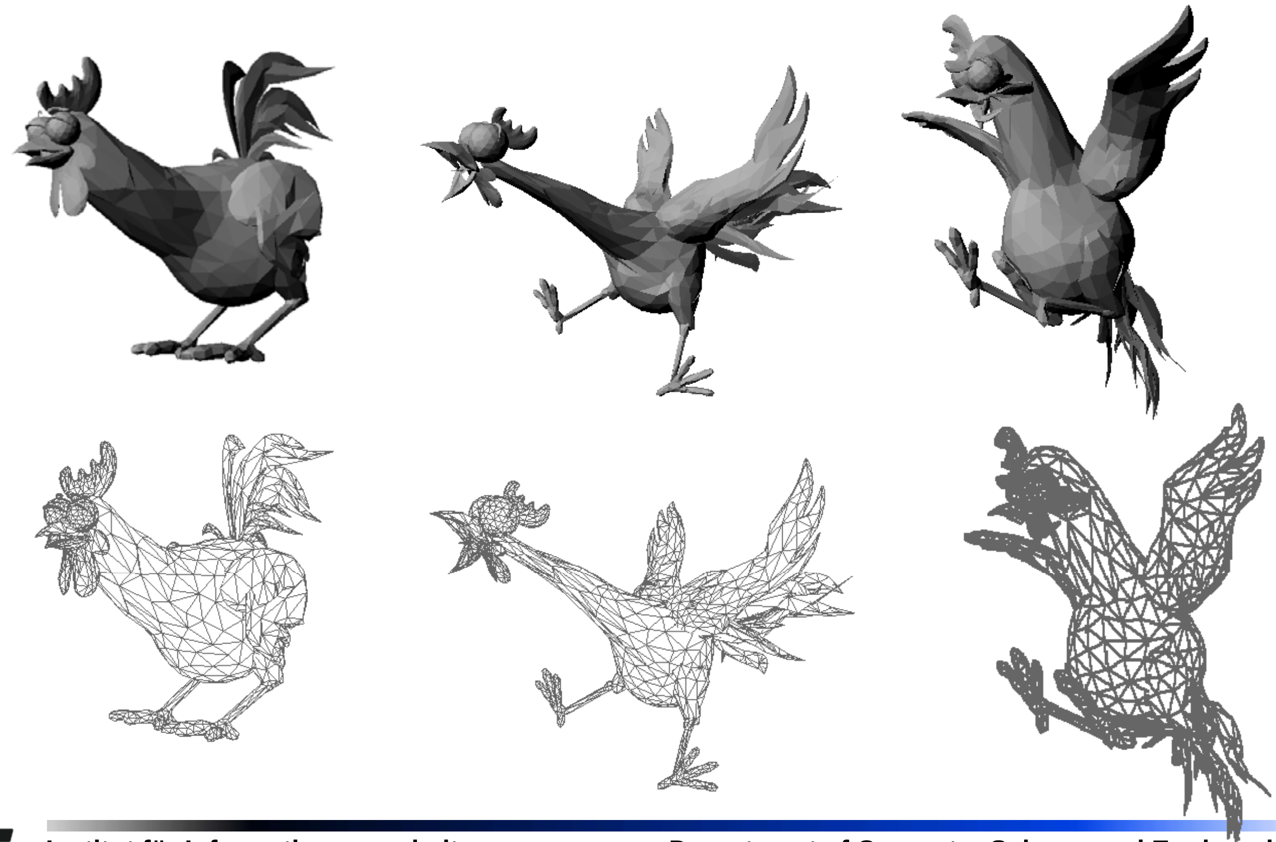
Introduction

Dynamic Mesh

Sequence of static 3D meshes

- with identical connectivity throughout the mesh sequence
- with changing geometry throughout the mesh sequence

Frame:	001	280	380

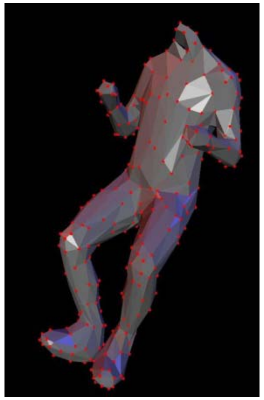


Introduction

- Raw representation of static meshes:

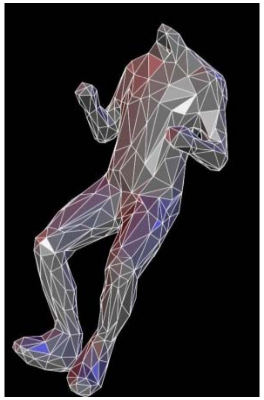
Geometry

(Vertex positions)



Connectivity

(Triangles)



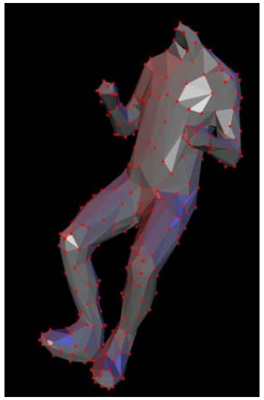
$v1$	0.5	0.34	1.45
$v2$	1.5	0.34	1.45
$v3$	0.7	0.54	3.35
$v4$	3.5	4.44	0.45
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t1$	1	2	3
$t2$	4	2	3
$t3$	4	3	5
$t4$	7	5	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Introduction

- Raw representation of static meshes:

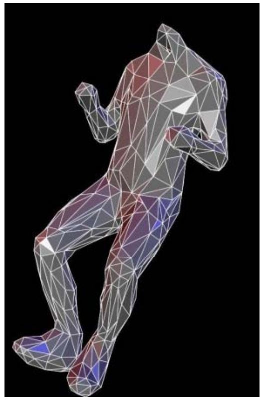
Geometry

(Vertex positions)



Connectivity

(Triangles)

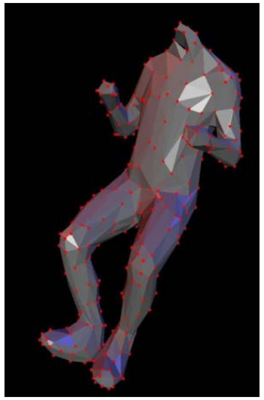


$v1$	0.5	0.34	1.45
$v2$	1.5	0.34	1.45
$v3$	0.7	0.54	3.35
$v4$	3.5	4.44	0.45
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t1$	1	2	3
$t2$	4	2	3
$t3$	4	3	5
$t4$	7	5	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

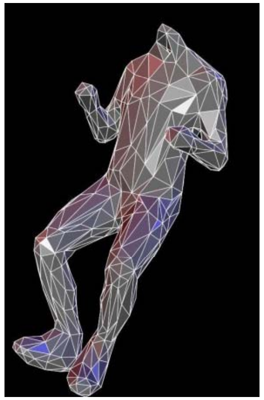
Introduction

- Raw representation of static meshes:

Geometry
(Vertex positions)



Connectivity
(Triangles)



$v1$	0.5	0.34	1.45
$v2$	1.5	0.34	1.45
$v3$	0.7	0.54	3.35
$v4$	3.5	4.44	0.45
⋮	⋮	⋮	⋮
$t1$	1	2	3
$t2$	4	2	3
$t3$	4	3	5
$t4$	7	5	1
⋮	⋮	⋮	⋮

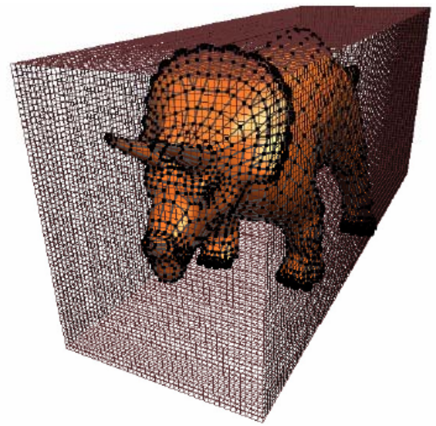
Introduction

- Raw bitrate of a static mesh:
  - Bit rate of **geometry data** given in bits per vertex [bpv]:
    - each vertex position *v1, v2, v3, …* is represented with 3 floats
    - lossless rate: 96 bpv

Geometry

$v1$	0.5	0.34	1.45
$v2$	1.5	0.34	1.45
$v3$	0.7	0.54	3.35
$v4$	3.5	4.44	0.45
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- quantization of 12 bits per component of each vertex position *v1, v2, v3*, … is 'treated' as lossless with regard to visual quality
- Lossy rate: 36 bpv



Introduction

- Raw bit rate of a static mesh:
  - Bit rate of **connectivity data**:
    - number of Vertices = V
    - number of Triangles = T
    - for each triangle *t1, t2, t3,* … we need
      $3\lceil \log_2 V \rceil$ bits per triangle
    - Due to the approximation $T \approx 2V$ (Euler-Equation) we obtain a bit rate of

      $6\lceil \log_2 V \rceil$ bpv

Connectivity

$t1$	1	2	3
$t2$	4	2	3
$t3$	4	3	5
$t4$	7	5	1
⋮	⋮	⋮	⋮

- total bit rate for **geometry** and **connectivity** data:

$$32 + 6\lceil \log_2 V \rceil \text{bpv}$$

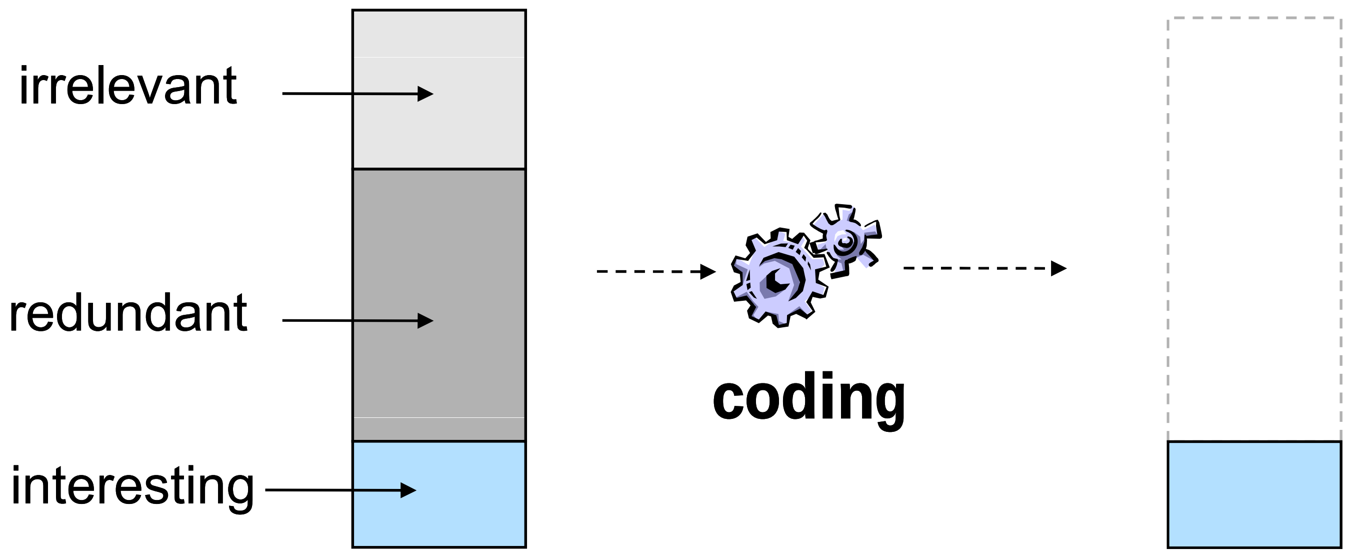
- bit rate dominated by connectivity

Introduction

Single-rate compression

Goal:

- remove irrelevant and redundant information present in the original description of the data

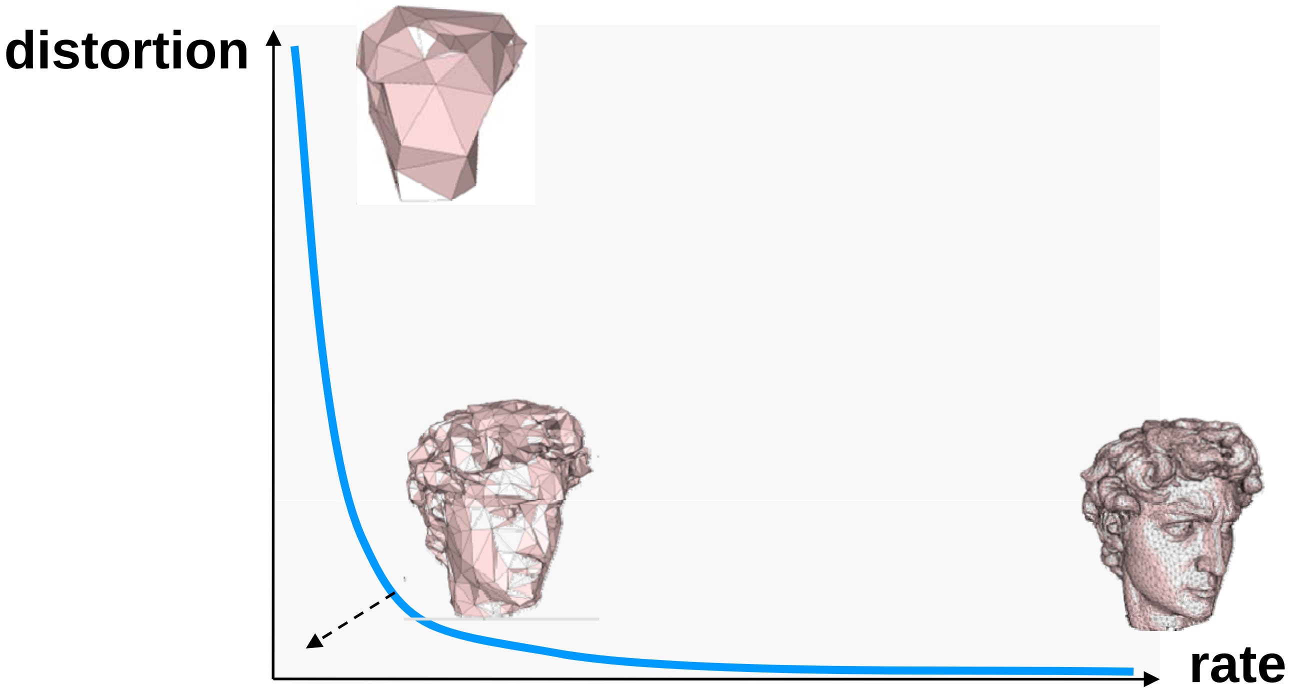


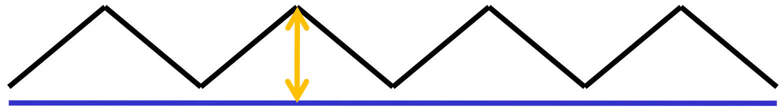
Scalable Compression Approaches

Scalable compression

Goal:

– optimize trade-off between rate and distortion



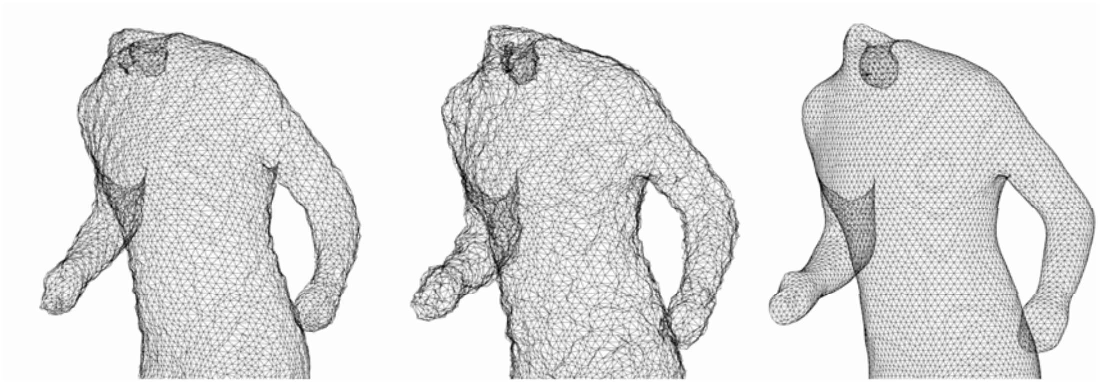


Distortion Measures

Distortion measures

- Key issue for R-D optimization
- Studied for static meshes
- Extended to dynamic meshes

- Problems
  - Specification
    - perception/absolute
  - Incorporation of time
  - Detection of artifacts
  - Invariances



  - Computational complexity (for RD optimizing coders)

Static mesh distortion measures

- Mean Squared Error (MSE)
- Image MSE
- Volume change
- Hausdorff distance

Mean squared error

$$MSE = \frac{1}{N}\sum_{i=1}^{n}\left(x_i - x_i'\right)^2$$

- Cannot deal with simplified meshes
- Only evaluates error at vertex positions
- Scale dependent value (can be normalized)
  - Sequence length
  - Coordinate span

Image MSE

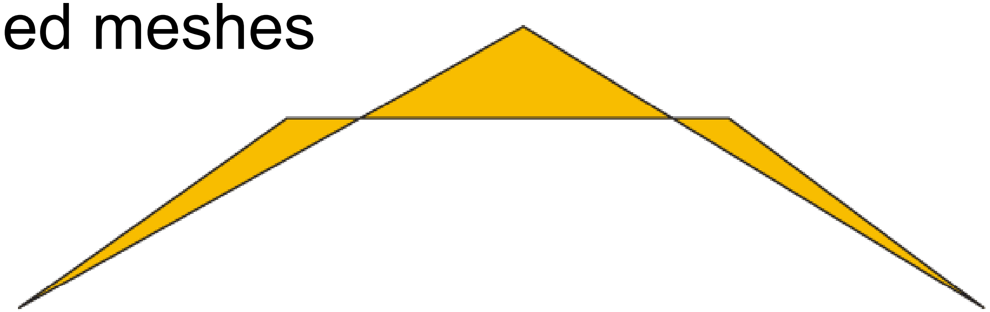
- Generate images from various viewpoints
- Compute MSE between pixel values
- Can deal with simplification
- Difficulties with viewpoint generation
- Questionable relevance of image MSE concept itself
  - Problems with artifacts
  - Insensitive to character of distortion

Peter Lindstrom, Gerg Turk. Image-driven simplification. ACM Transactions on Graphics. 19(3):204-241, 2000.

Volume change

- Applied locally during simplification
- Zero total change can be reached
- Usually combined with minimizing the volume „between“

- Quick to evaluate
- Can deal with simplified meshes



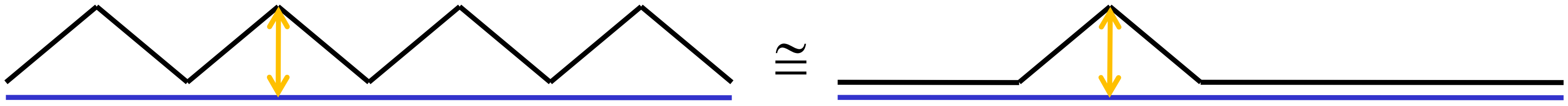
Peter Lindstrom and Greg Turk. Fast and memory efficient polygonal simplification. In *IEEE Visualization, pages 279-286, 1998.*

Hausdorff distance

- Search for „biggest problem“
- From each point of surface A find the closest point on surface B
- From each point of surface B find the closest point on surface A
  - including interior points of triangles (usually only sampled)
- Determine the largest of these distances
- Can deal with simplified meshes

- Some public implemetations (MESH, Metro)

Hausdorff Distance problems

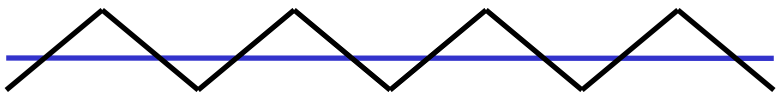


- Is this what we want? (sometimes may be)

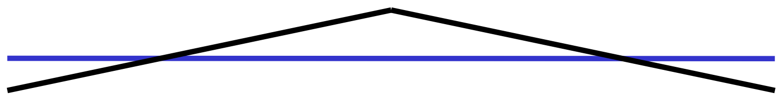
- Is this what we want?

Hausdorff RMS

$$\sqrt{\frac{1}{N}\sum_{x\in\rho_1}\sum_{y\in\rho_2}(x-y)^2}$$



$\cong$



Dynamic mesh distortion measures

- KG error
- 4D Hausdorff
- DA error (Ribbon error)

KG error

- Dynamic mesh represented by a matrix A:
- Mean values of each column in matrix E(A):

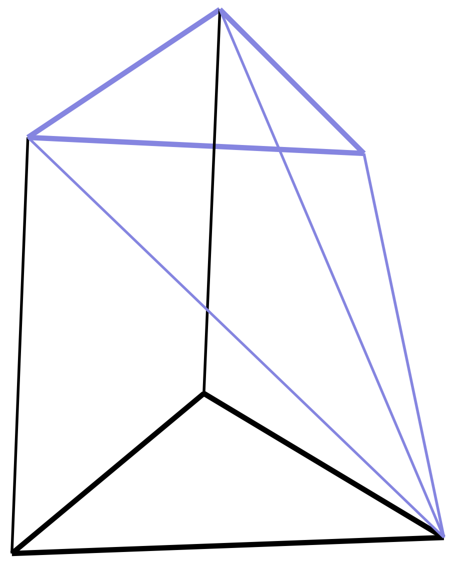
$$A=\begin{pmatrix} x_1^1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 & x_1^6 & \cdots \\ y_1^1 & y_1^2 & y_1^3 & y_1^4 & y_1^5 & y_1^6 & \cdots \\ z_1^1 & z_1^2 & z_1^3 & z_1^4 & z_1^5 & z_1^6 & \cdots \\ x_2^1 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & x_2^6 & \cdots \\ y_2^1 & y_2^2 & y_2^3 & y_2^4 & y_2^5 & y_2^6 & \cdots \\ z_2^1 & z_2^2 & z_2^3 & z_2^4 & z_2^5 & z_2^6 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \qquad E(A)=\begin{pmatrix} \overline{x^1} & \overline{x^2} & \overline{x^3} & \overline{x^4} & \overline{x^5} & \overline{x^6} & \cdots \\ \overline{y^1} & \overline{y^2} & \overline{y^3} & \overline{y^4} & \overline{y^5} & \overline{y^6} & \cdots \\ \overline{z^1} & \overline{z^2} & \overline{z^3} & \overline{z^4} & \overline{z^5} & \overline{z^6} & \cdots \\ \overline{x^1} & \overline{x^2} & \overline{x^3} & \overline{x^4} & \overline{x^5} & \overline{x^6} & \cdots \\ \overline{y^1} & \overline{y^2} & \overline{y^3} & \overline{y^4} & \overline{y^5} & \overline{y^6} & \cdots \\ \overline{z^1} & \overline{z^2} & \overline{z^3} & \overline{z^4} & \overline{z^5} & \overline{z^6} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$E_{KG} = 100 \frac{\left\| A - A' \right\|}{\left\| A - E(A) \right\|}$$

Zachi Karni and Craig Gotsman. Compression of soft-body animation sequences. In *Computers & Graphics, volume 28, pages 25-34, 2004.*

4D Hausdorff

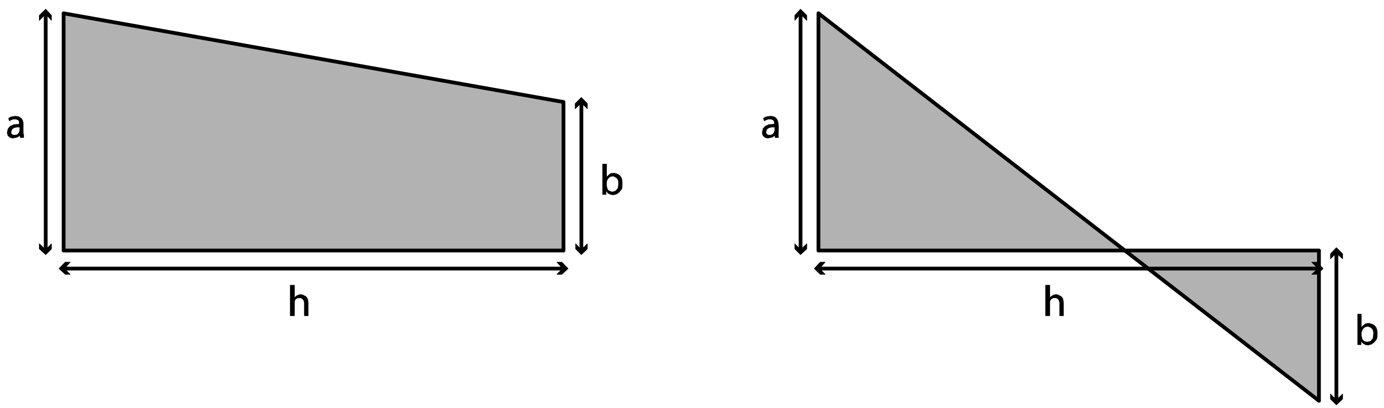
- Extension of Hausdorff distance
- 4D tetrahedral mesh representation
- 4D euclidean distance
  - Space/time unit relation problem
- Slow to evaluate
- Can deal with different framerates



Libor Váša and Václav Skala. A spatio-temporal metric for dynamic mesh comparison. In *AMDO Lecture Notes on Computer Graphics, 4096, pages 29-37. Springer*-Verlag Berlin Heidelberg, 2006.

DA Ribbon error

- Constructed for dynamic meshes
- Ribbons formed by original and moved vertex in two subsequent frames

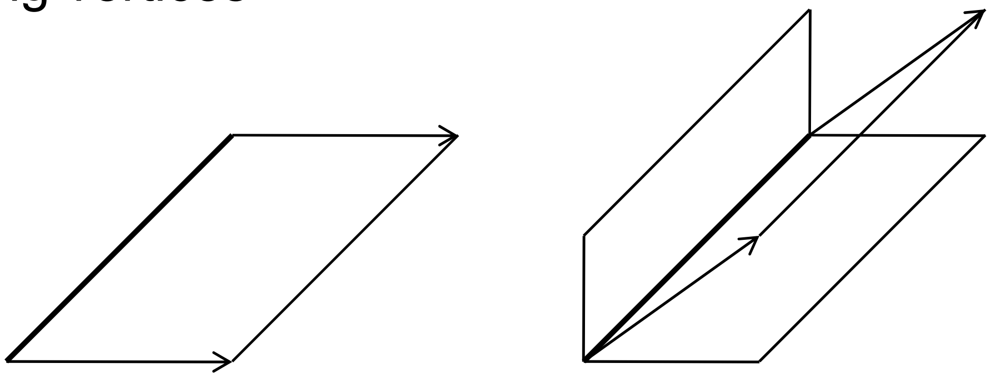


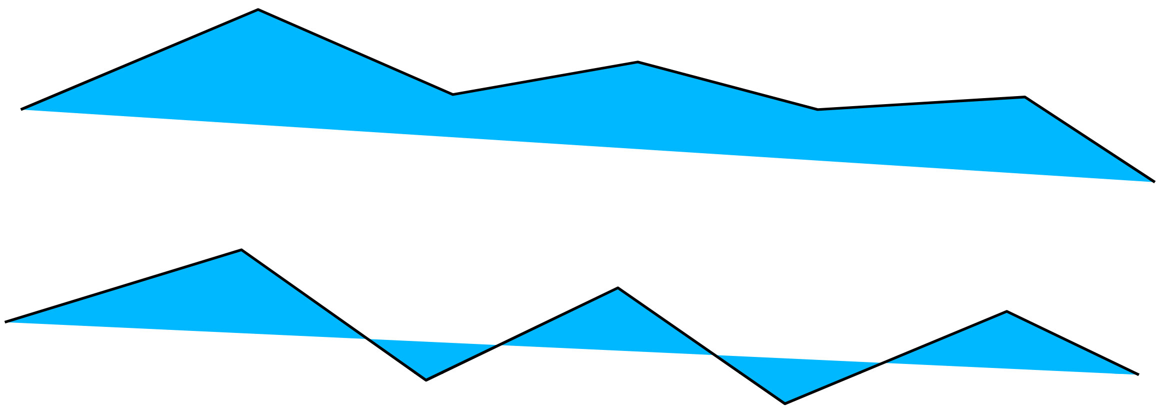
$$A = t\frac{a+b}{2} \qquad A_{tw} = t\frac{a^2+b^2}{2(a+b)}$$

- Error = sum over all vertices in all frames

Problems of Da error

- Not rotation invariant (evaluated separately for each axis!)
- Questionable treatment of temporal artifacts
  - Favors oscilating vertices





Measures summary

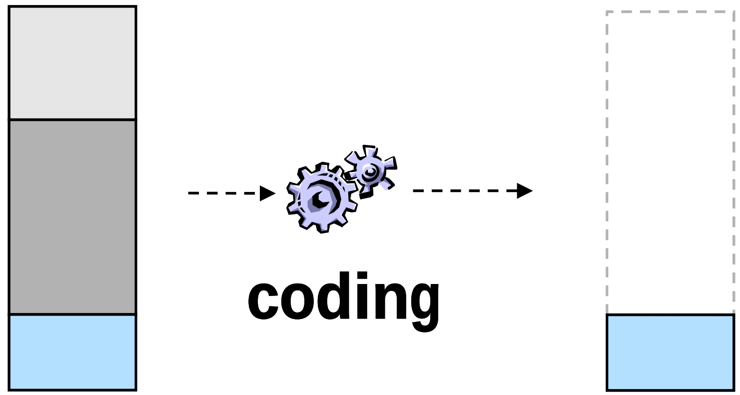
	MSE	HD	HD sum	KG	4D HD	DA
dynamic	no	no	yes	yes	yes	yes
deals with simplification	no	yes	yes	no	yes	no
Deals with varying framerate	no	no	no	no	yes	no
fast	yes	no	no	yes	no	yes
rotation invariant	yes	yes	yes	yes	yes	no
perception based	no	no	No	no	no	no

Measures summary

- Many tools available
- No consensus

- All the measures fail when more than one type of distortion is introduced

Compression of Static 3D Meshes



Single-Rate
Compression Approaches

Single-Rate Compression Approaches

Single rate compression

- restore model ***after*** decoding
  - aim: download and decompress
- compression process driven by connectivity
  - optimized for compressing connectivity

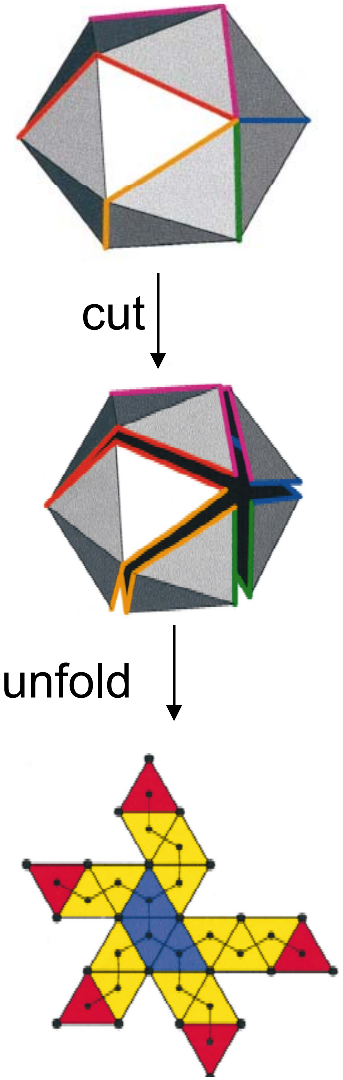
Compression approaches

- approaches for connectivity compression already matured
  - Topological surgery [Taubin ’98 MPEG-4]
  - Cut-border Machine [Gumhold & Strasser ‘98]
  - Triangle Mesh Compression [Touma & Gotsman ‘98]
  - Edgebreaker and derivatives [Rossignac et al. ‘99]
  - Valence-Driven Connectivity Encoding for 3D Meshes [Alliez & Desbrun ‘01]

Single-Rate Compression Approaches

Topological Surgery [Taubin '98]

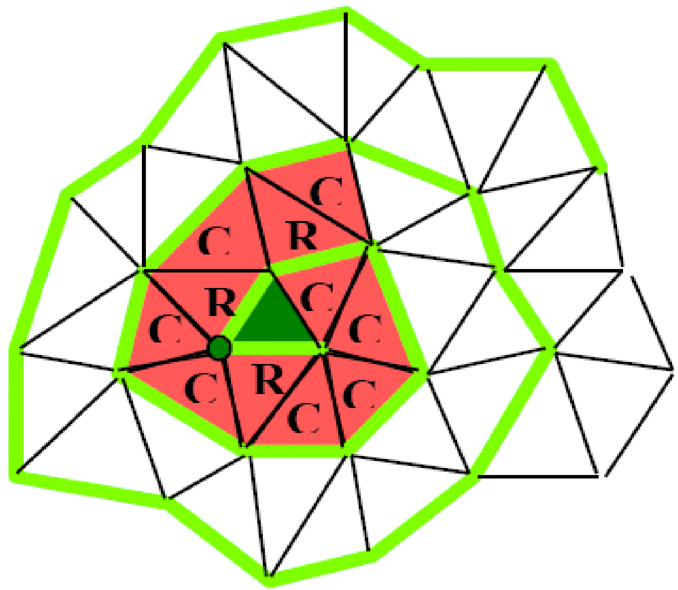
- Cut the mesh and unfold it on a plane
- Encode the unfolded representation of the mesh
- Encode the "cuts" needed to reconstruct the 3D mesh from the unfolded representation
- several passes required for encoding/decoding
- Part of MPEG-4 standard
- Bit-rate around 4 bpv



Single-Rate Compression Approaches

Edgebreaker [Rossignac et al. ‘99]

- Traverse triangles
  - encode spirals of triangles
- Codebook of 5 symbols: CLRES
  - describes all possible configurations per triangle
  - 1 symbol per triangle to encode

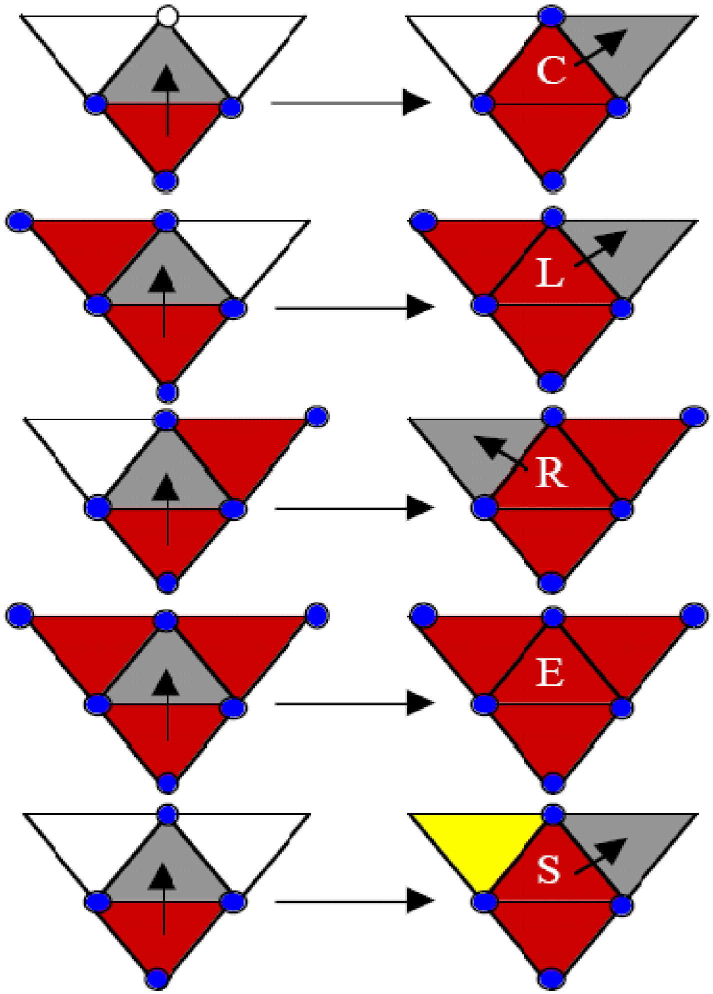


[CCCCRCCRCRC…]

Single-Rate Compression Approaches

Edgebreaker [Rossignac et al. '99]

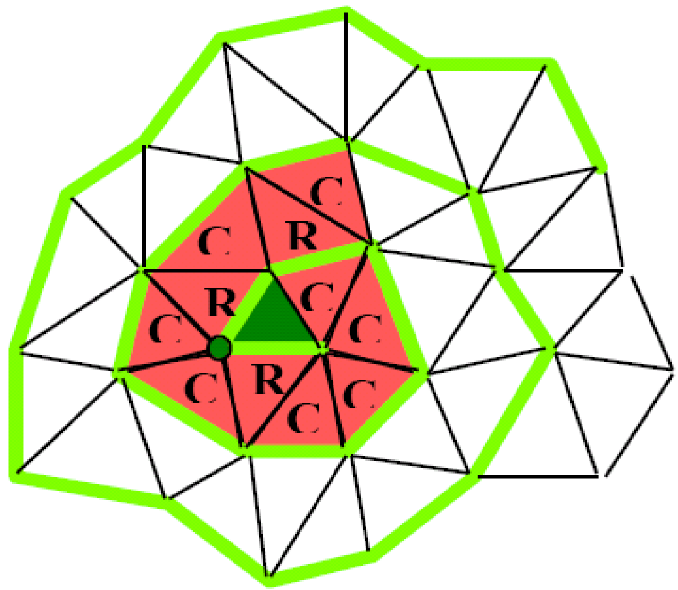
- Traverse triangles
  - encode spirals of triangles
- Codebook of 5 symbols: CLRES
  - describes all possible configurations per triangle
  - 1 symbol per triangle to encode



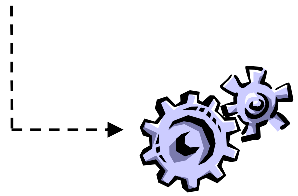
Single-Rate Compression Approaches

Edgebreaker [Rossignac et al. '99]

- Traverse triangle
  – encode spirals of triangles
- Codebook of 5 symbols: CLRES
  – describes all possible configurations per triangle
  – 1 symbol per triangle to encode
    - C=0 (50 % of all symbols)
    - L=100
    - R=101
    - E=110
    - S=111
- Upper bound for connectivity: 2bpt = 4bpv



[CCCCRCCRCRC…]



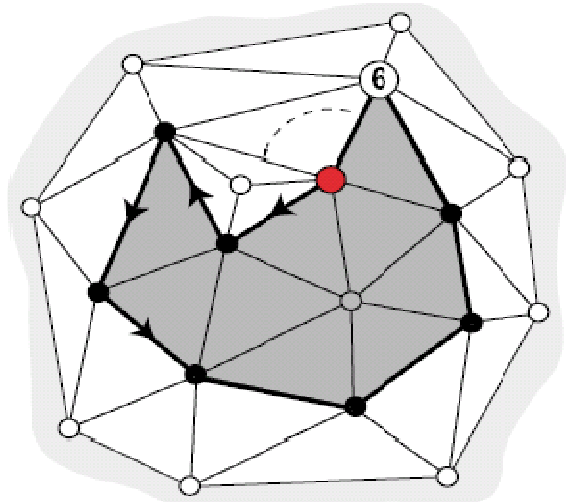
[00001010...]

Single-Rate Compression Approaches

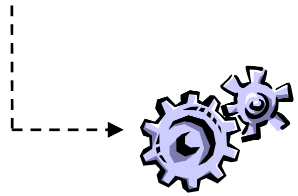
Valence Driven Compression

[Touma & Gotsman ’98, Alliez & Desbrun ‘01]

- Traverse vertex based
  - conquer edges of pivot vertices
  - generate valence code



[5566…]



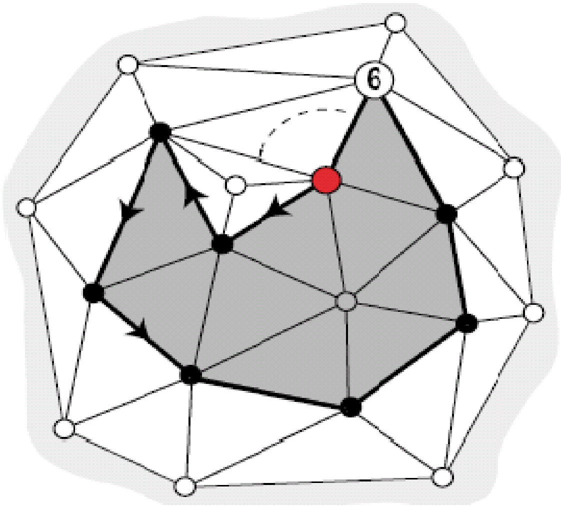
[00001010...]

Single-Rate Compression Approaches

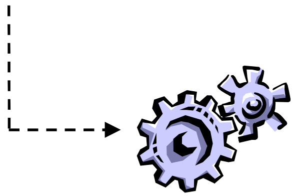
Valence Driven Compression

[Touma & Gotsman ’98, Alliez & Desbrun ‘01]

- Traverse vertex based
  - conquer edges of pivot vertices
  - generate valence code
- Entropy coding of valences
  - large meshes: average vertex valence is 6
  - regular meshes: all vertex valences are 6
- Approach benefits from regularity
- Asymptotically optimal
  - Upper bound of approx. 3.25 bpv for large meshes
- In practice: 2.0 - 3.5 bpv



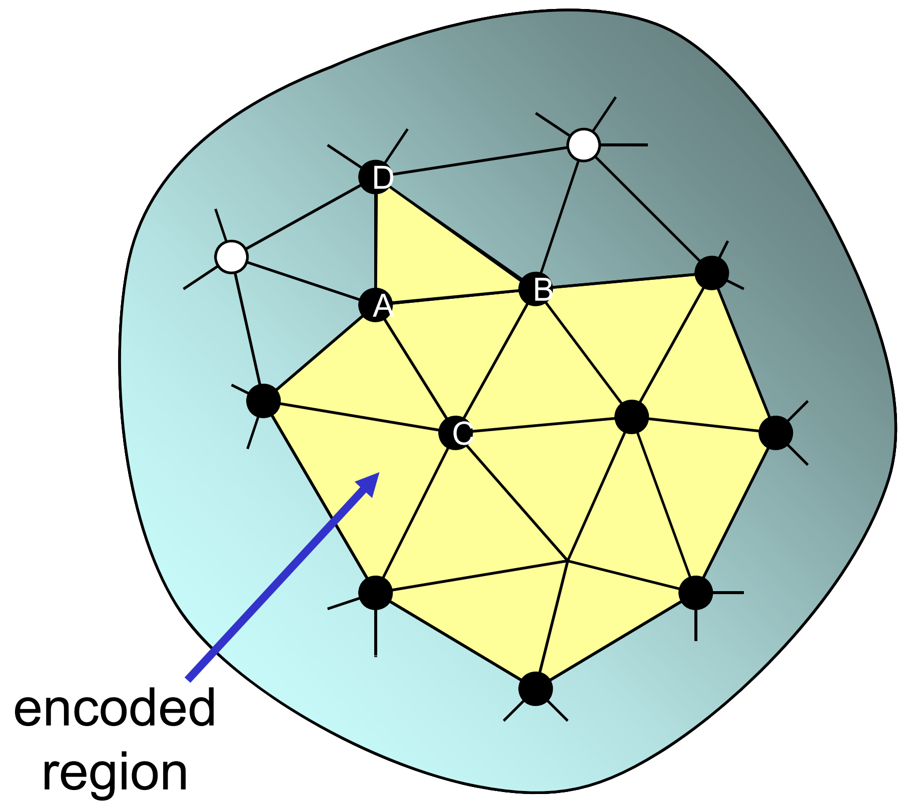
[5566…]



[00001010...]

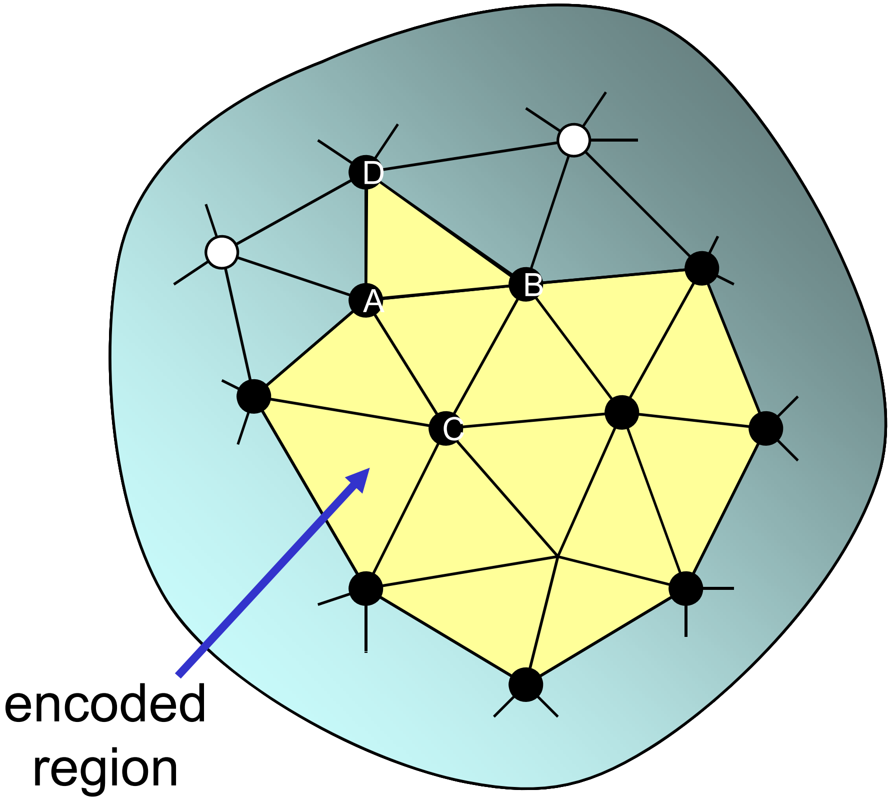
Single-Rate Compression Approaches

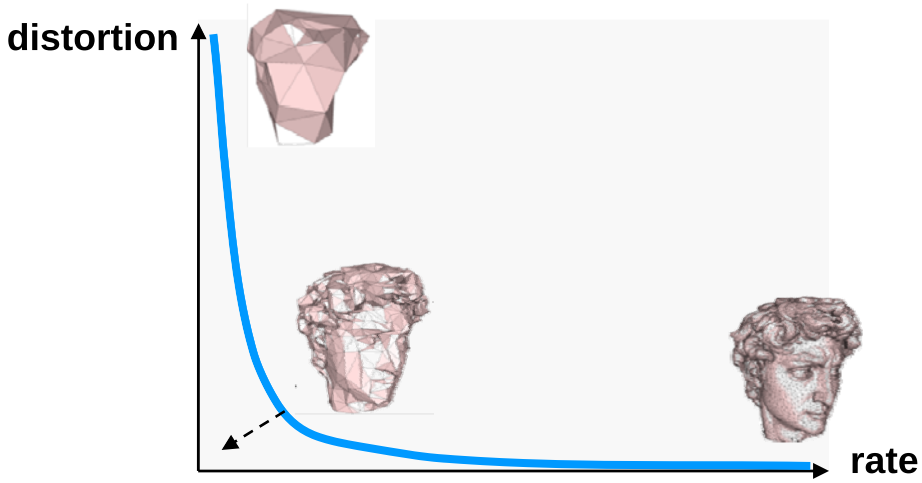
- Predictive geometry compression:
  - linear predictors exploiting encoded vertex positions of the current neighborhood
  - parallelogram predictor
    - P(D)=A+B-C
  - encode error: E=D-P(D)



Single-Rate Compression Approaches

- Predictive geometry compression:
  - linear predictors exploiting encoded vertex positions of the current neighborhood
  - parallelogram predictor
    - P(D)=A+B-C
  - encode error: E=D-P(D)
- Bit-rates in practice
  - 12 to 20 bpv



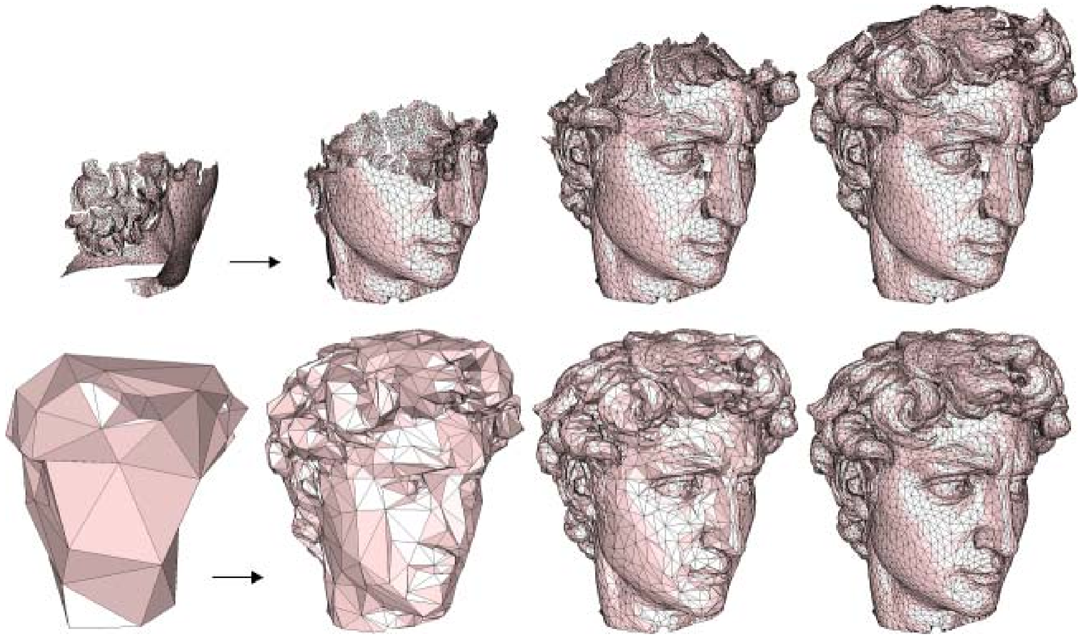


Scalable
Compression Approaches

Scalable Compression Approaches

- Scalable compression
  - restore model successively
    - aim: get a first impression of the object by decoding a small part of the bit stream and increase object's quality by decoding more data

Single-rate *vs.* scalable compression

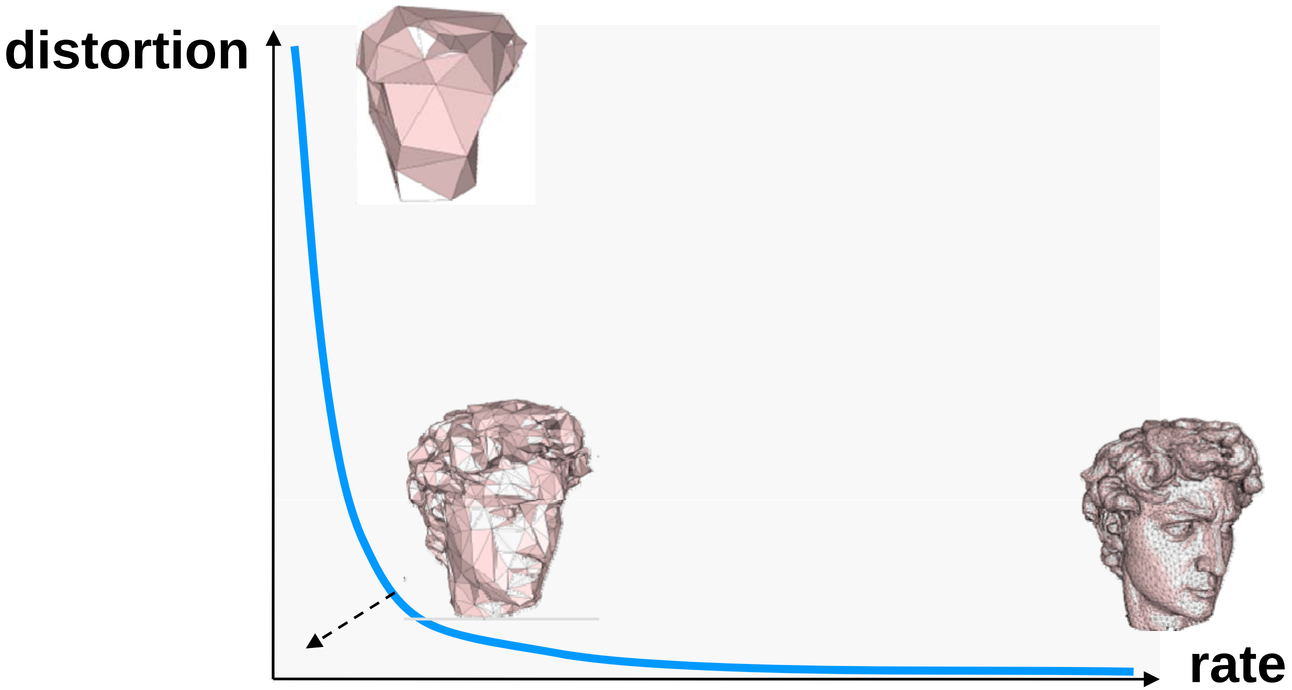


Scalable Compression Approaches

Scalable compression

Goal:

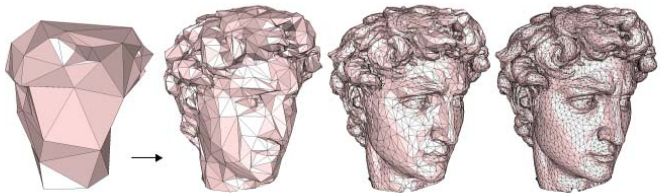
– optimize trade-off between rate and distortion



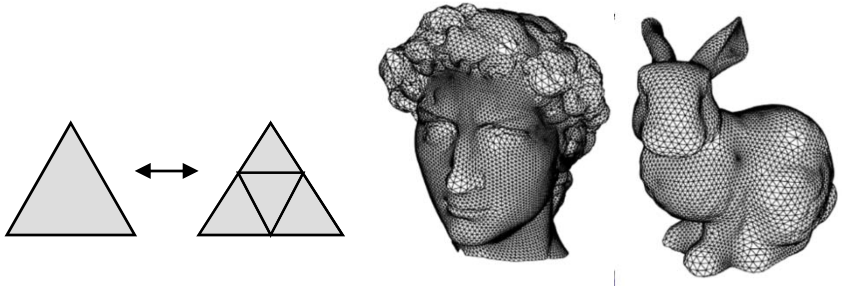
Scalable Compression Approaches

Types of Scalable Compression:

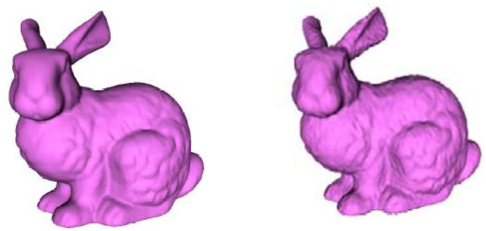
- Spatially scalable
  - Successive increase in the number of vertices (i.e. resolution)
    - Progressive meshes



    - Semi-regular meshes



- Quality scalable
  - Successive increase in accuracy of vertex positions (number of vertices is fixed)
    - Global spectral transforms (PCA, Laplace)



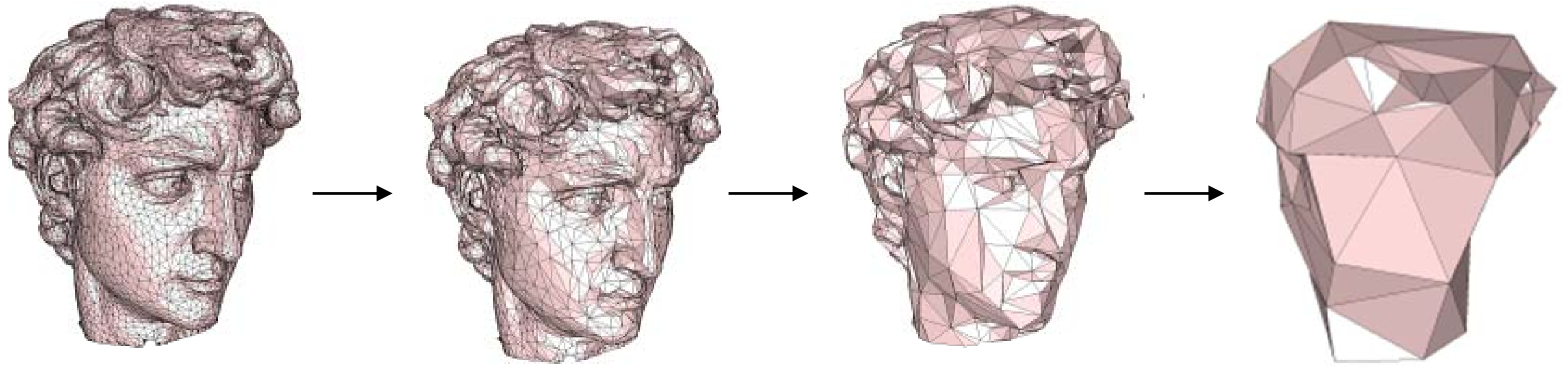
Progressive Mesh Based Compression

Compression approaches based on Progressive Meshes (PM)

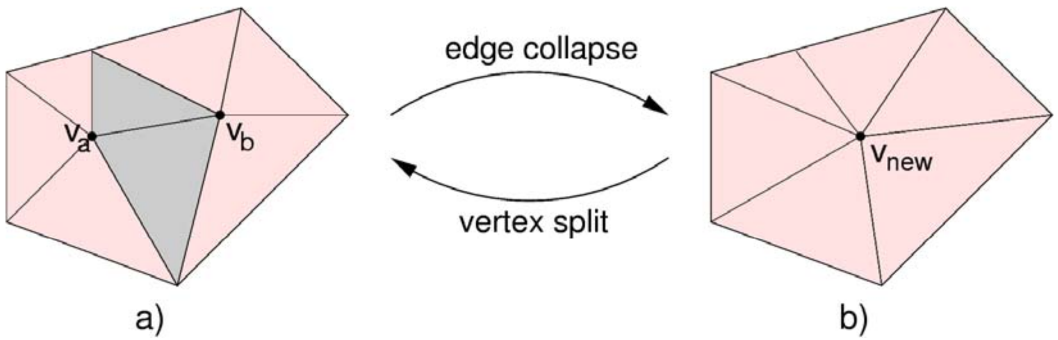
- approaches already matured
- encoding process driven by connectivity
  - Progressive Meshes [Hoppe ‘96]
  - Progressive Forest Split Compression [Taubin et al. ’98, MPEG-4]
  - Progressive Compression of Arbitrary Triangular Meshes [Cohen-Or et al. ’99]
  - Progressive Compression and Transmission of Arbitrary Triangular Meshes [Bajaj et al., ’99]
  - Compressed Progressive Meshes [Pajarola & Rossignac ‘00]
  - Progressive Compression for Lossless Transmission of Triangle Meshes [Alliez, Desbrun, ‘01]

Progressive Mesh Based Compression

- Basic Idea
  - simplify a mesh using a sequence of simplification operations



  - most popular simplification operation: edge collapse



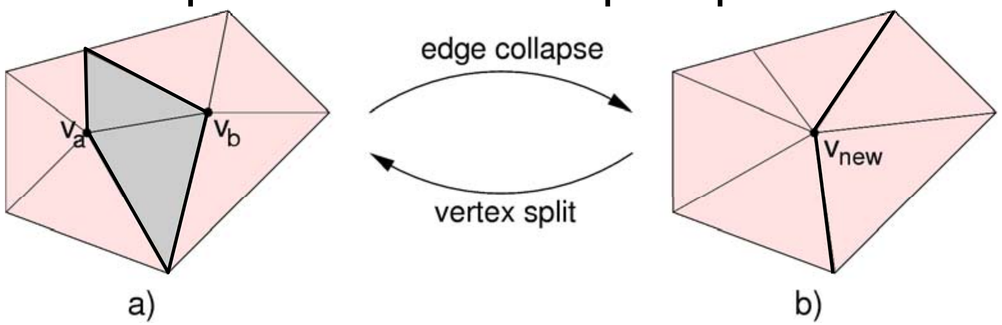
  - invert simplification process during encoding (coarse to fine)

Progressive Mesh Based Compression

- General encoding approach for PM:
  - encode **where** a vertex (or a set of vertices) is removed
  - encode **how** this vertex (or set of vertices) is removed
  - encode predictively the corresponding vertex position (positions)

Progressive Mesh Based Compression

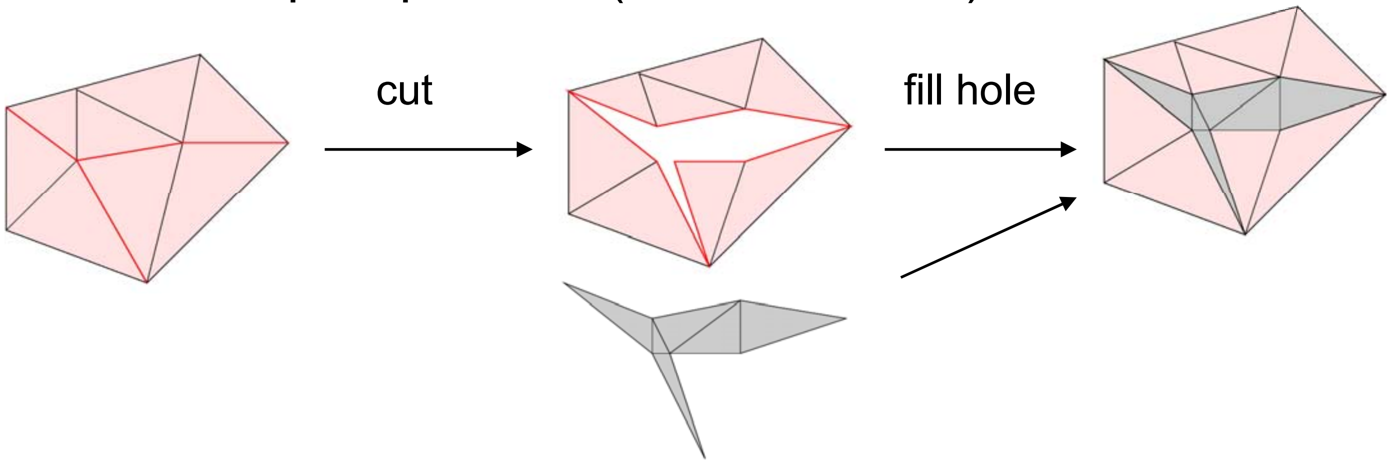
- PM [Hoppe '96]
  - encode sequence of vertex split operations



  - about 15 bpv for each simplification operation
  - delta encode vertex position
    - 30-50 bpv
  - requires encoding of random access to vertices
    - very costly
  - but provides high flexibility
    - very smooth transition from coarse to fine due to refinement on vertex basis (vertex granularity)

Progressive Mesh Based Compression

- Progressive Forest Split [Taubin et al. ’98]
  - forest split operation (at the decoder):



  - encode cuts and internal triangulations
    - 10 bpv
  - encode predictively vertex positions
    - about 30 bpv
  - approach is part of MPEG-4 standard

Progressive Mesh Based Compression

- Compressed Progressive Meshes (CPM) [Pajarola et al. ’00]
  - encode vertex split operations in batches
    - encoding in batches is more efficiently compared to encoding single operations (cp. with Hoppe’s approach)
    - encoding is more efficiently at the expense of reduced granularity
      - each batch removes about 30% of vertices
  - encode batches of simplification operations
    - about 7 bpv
  - encode predictively vertex positions using generalized butterfly scheme
    - about 15 bpv
  - 50 % increase in compression efficiency compared to MPEG-4 at the expense of reduced granularity

Scalable Compression Approaches

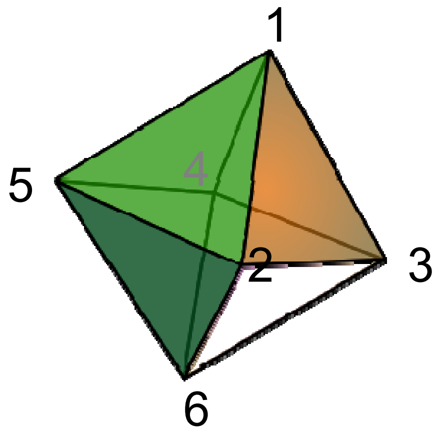
Spectral Compression Approaches

- separated connectivity and geometry compression for arbitrary meshes using spectral transforms
  - **Spectral Compression of Mesh Geometry** [Karni & Gotsman ‘00]
  - **3D Mesh Compression using Fixed Spectral Basis** [Karni & Gotsman ‘00]

- Basic idea
  - encode connectivity using an arbitrary single resolution approach
  - derive a transform from connectivity
  - transform the 3 V-dimensional vectors X, Y, Z which represent mesh geometry
  - encode transform coefficients

Progressive Mesh Based Compression

- Spectral Compression [Karni & Gotsman ‘00]
  - calculate graph Laplacian L from connectivity



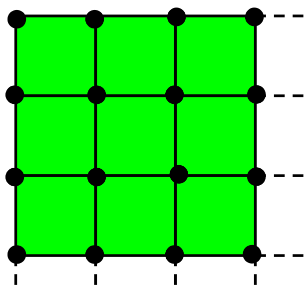
adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \qquad L = I - \tfrac{1}{4} A$$

  - derive transform form graph Laplacian L
    - eigenvectors of L create transform matrix T
  - eigenvectors are ranked according their respective eigenvalues
    - eigenvectors correspond to basis functions in Fourier transform
    - eigenvalues correspond to frequencies in Fourier transform

Progressive Mesh Based Compression

- Spectral Compression [Karni & Gotsman ‘00]
  - special case: regular grids
  - calculate graph Laplacian L from regular grid



$$L = I - \tfrac{1}{4} A$$

  - eigenvectors of L **are equal** to 2D-Fourier basis functions
  - graph Laplacians allow to generalize Fourier transform to data sampled on irregular grids

Progressive Mesh Based Compression

- Spectral Compression [Karni & Gotsman ‘00]
  - encode transformed vectors TX, TY, and TZ
  - gain in bit rate in geometry compression of over 50 % compared to single resolution approaches
  - problem of this approach
    - eigenvector decomposition of Laplacian L (size V x V) is required at the encoder **and** decoder
      - computationally very demanding
    - several passes over the whole mesh needed to calculate transformed geometry (global transform)
  - approach is rather of theoretical interest

Scalable Compression Approaches

Compression Approaches Based on Semi-Regular Meshes

- efficient for high-resolution meshes (e.g. from range scanning)
- consider 3D meshes as one instance of surface geometry
- connectivity gives additional degrees of freedom
  - MAPS: multiresolution adaptive parameterization of surfaces [Lee et al. '98]
  - Progressive Geometry Compression [Khodakovsky et al. '00]
  - Normal Meshes [Guskov et al. '00]
  - Compression of Normal Meshes [Kodakovsky & Guskov '03]

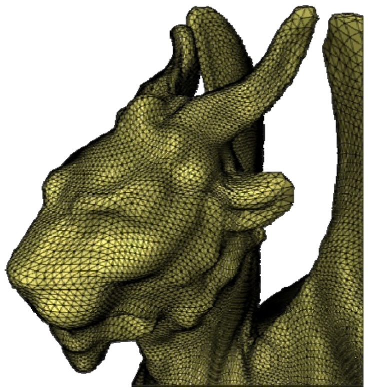
- Basic idea
  - create a semi regular mesh (a new instance of the 3D mesh)
    - semi regular mesh exhibits a multi-resolution hierarchy
  - encode geometry predictively from coarse to fine
    - use coarse representation as domain for prediction of details
      - employ subdivision schemes as predictors

Semi-Regular Meshes Based Approaches

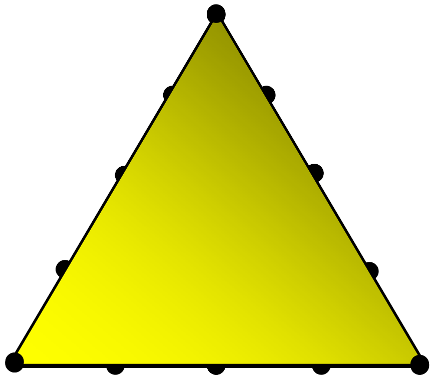
irregular mesh

remesh

semi-regular mesh



subsampling of
semi-regular mesh

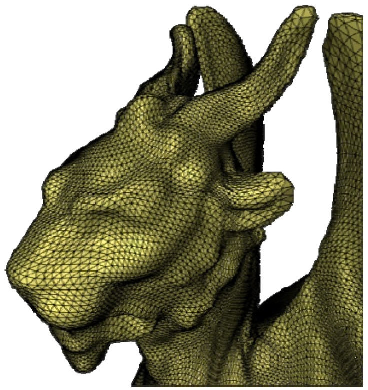


Semi-Regular Meshes Based Approaches

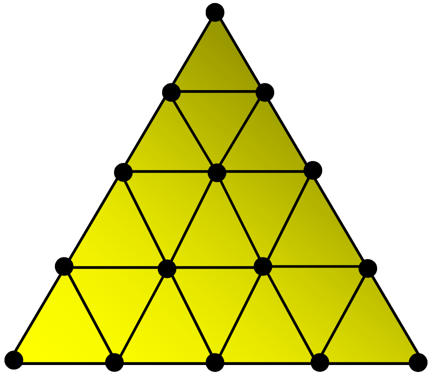
irregular mesh

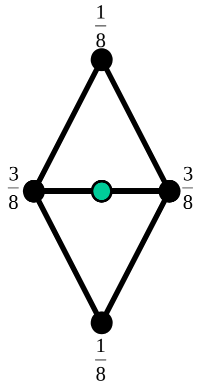
remesh →

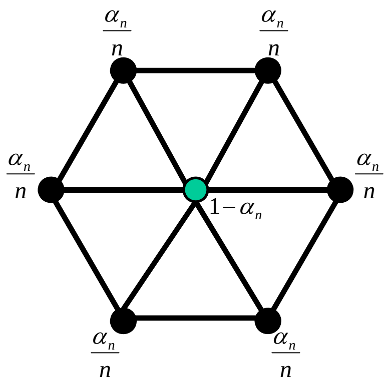
semi-regular mesh



predict from
coarse to fine







- connectivity refinement based on edge bisection
- use subdivision scheme (e.g. Loop, Butterfly) for prediction of vertex positions from coarse to fine mesh

$$\alpha_n = \frac{1}{64}\left(40 - \left(3 + 2\cos\left(\frac{2\pi}{n}\right)\right)^2\right)$$

Semi-Regular Meshes Based Approaches

Progressive Geometry Compression [Khodakovsky et al. ‘00]

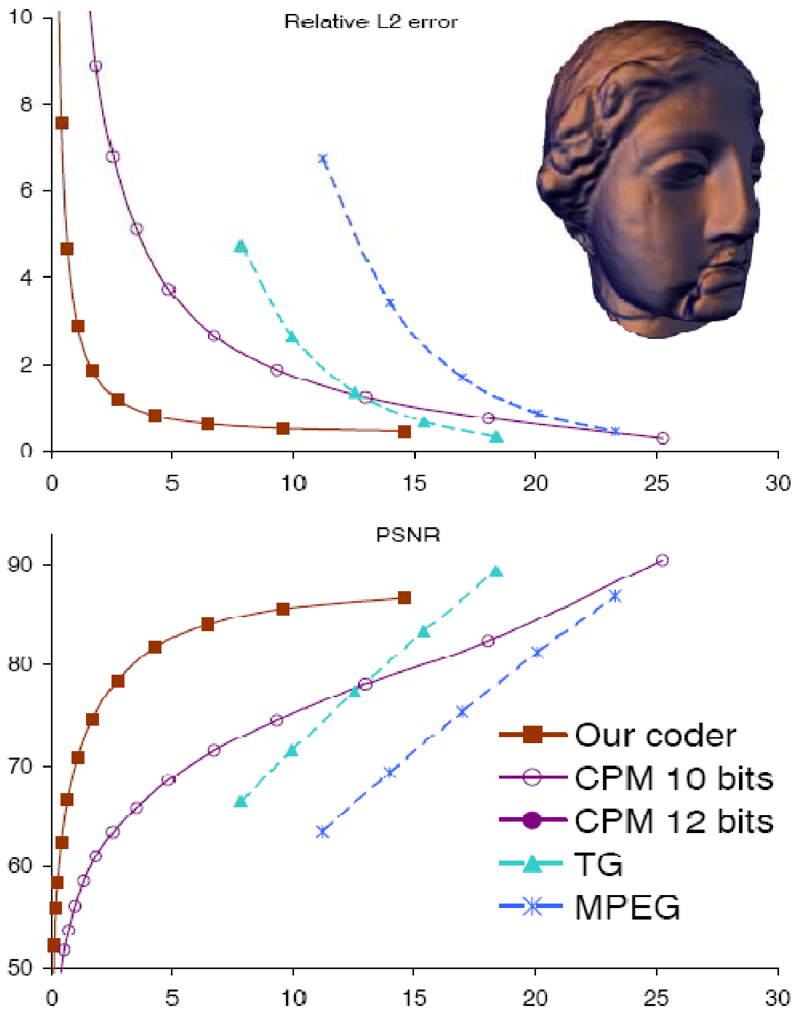
- remeshing: irregular to semi-regular mesh
  - remeshing based on MAPS algorithm [Lee et al. ‘98]
- encode base mesh using single resolution approach
- encode number of subdivision levels
- encode vertex positions predictively
  - employ predictor based on Loop subdivision
  - encode prediction errors (3D vectors) using zerotrees

Compression of Normal Meshes [Kodakovsky & Guskov ‘03]

- same as in Progressive Geometry compression, but
  - use special normal remesher [Guskov et al. ‘00]
  - prediction errors result in scalar offsets in normal direction
    - encode 1D instead of 3D prediction errors

Semi-Regular Meshes Based Approaches

RD-curves: „Progressive Geometry Compression„



Progressive Geometry Compression

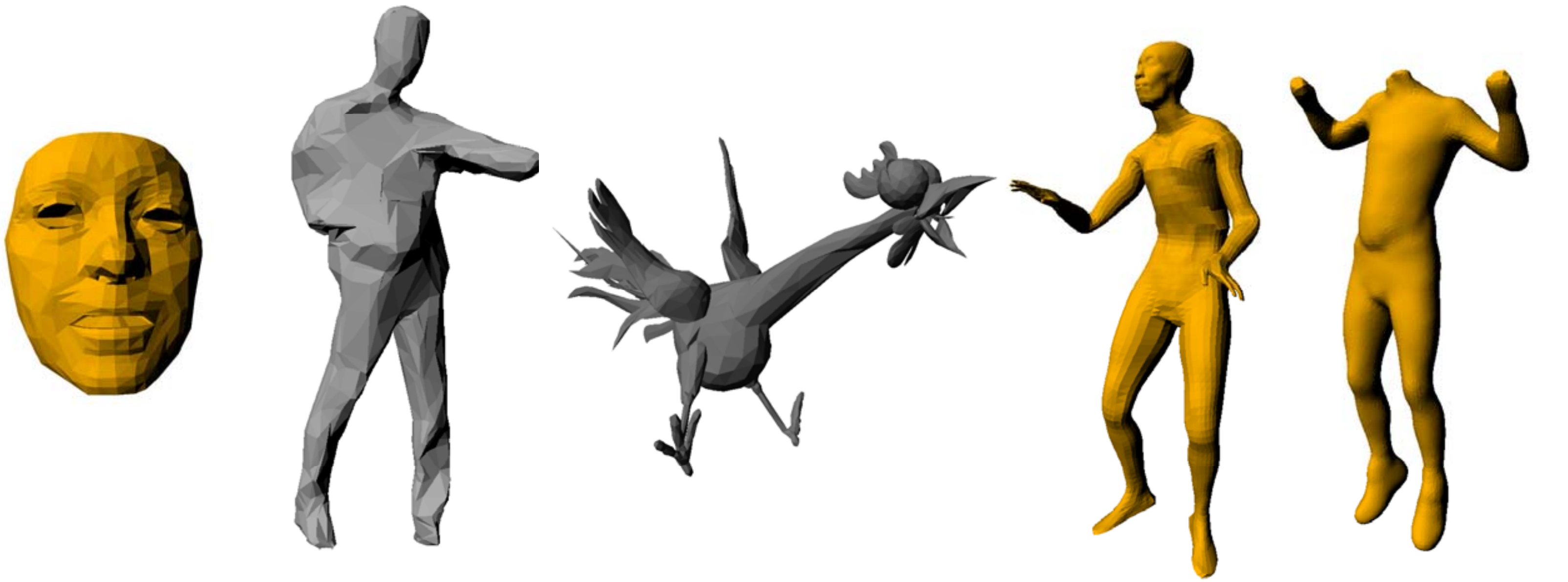
- gain of 12 dB compared to CPM

Compression of Normal Meshes

- improves 2-5dB compared to "Progressive Geometry Compression"

Summary and Conclusions

- Static mesh compression matured
  - raw bit rate dominated by connectivity: $32 + 6\lceil \log_2 V \rceil$
  - theoretical bounds for connectivity compression achieved
    - upper bound 4 bpv (bound independent of V)
      - with Edgebreaker
    - asymptotic (for large meshes) upper bound 3.245 bpv
      - with valence based compression
  - connectivity guided geometry compression: 12-20 bpv
  - separation of connectivity and geometry compression leads to significant gains in geometry compression
    - spectral compression: gain of 50% (of theoretical interest)
  - semi-regular meshes
    - very efficient for densely sampled meshes: 5-10 bpv
    - but encoding is computationally demanding
      - remeshing required



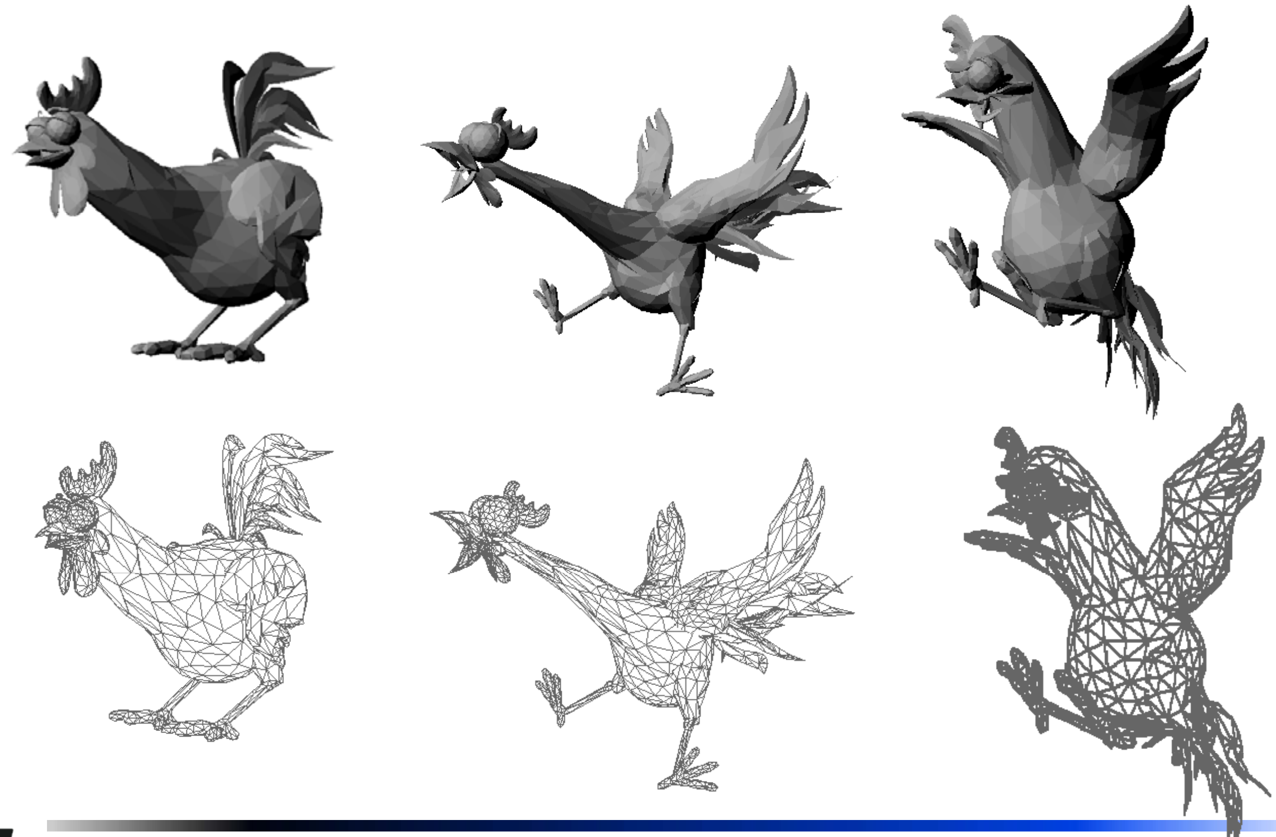
Compression of Dynamic 3D Meshes

Compression of Dynamic 3D Meshes

Dynamic Mesh

Sequence of static 3D meshes
- with identical connectivity throughout the mesh sequence
- with changing geometry throughout the mesh sequence

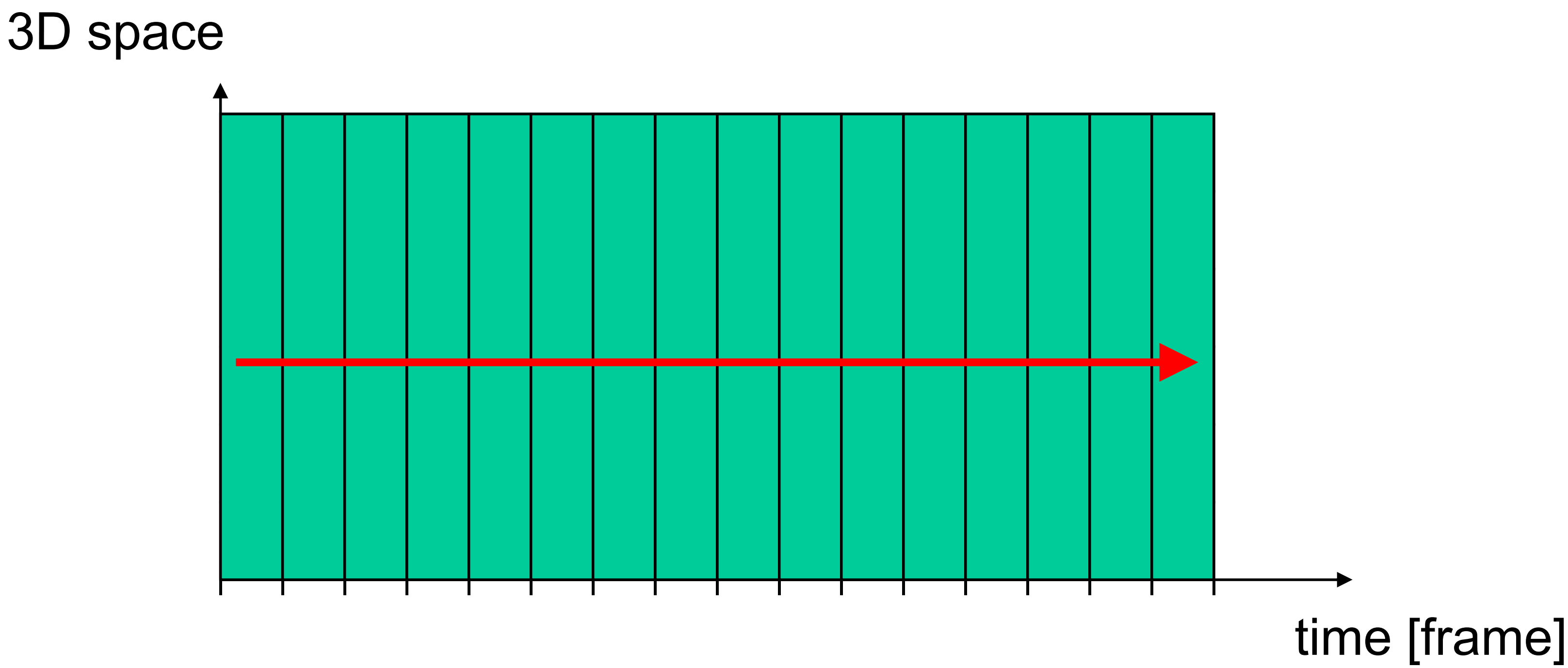
Frame:	001	280	380



Scalable Compression Types

Single resolution (no scalability)

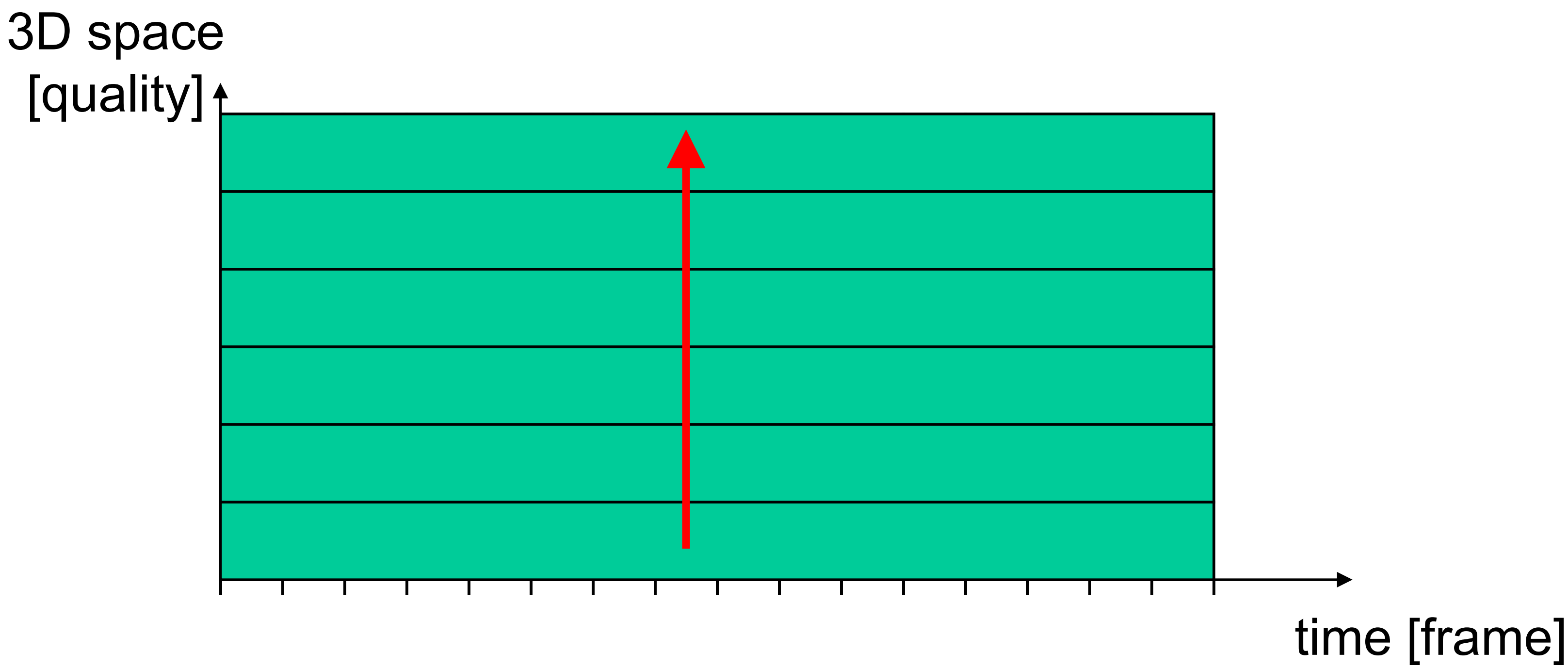
– frame-wise encoding/decoding



Scalable Compression Types

Quality scalability on animation level

- successive decoding increases the accuracy of all vertex positions of the entire animation
- no frame-wise decoding possible

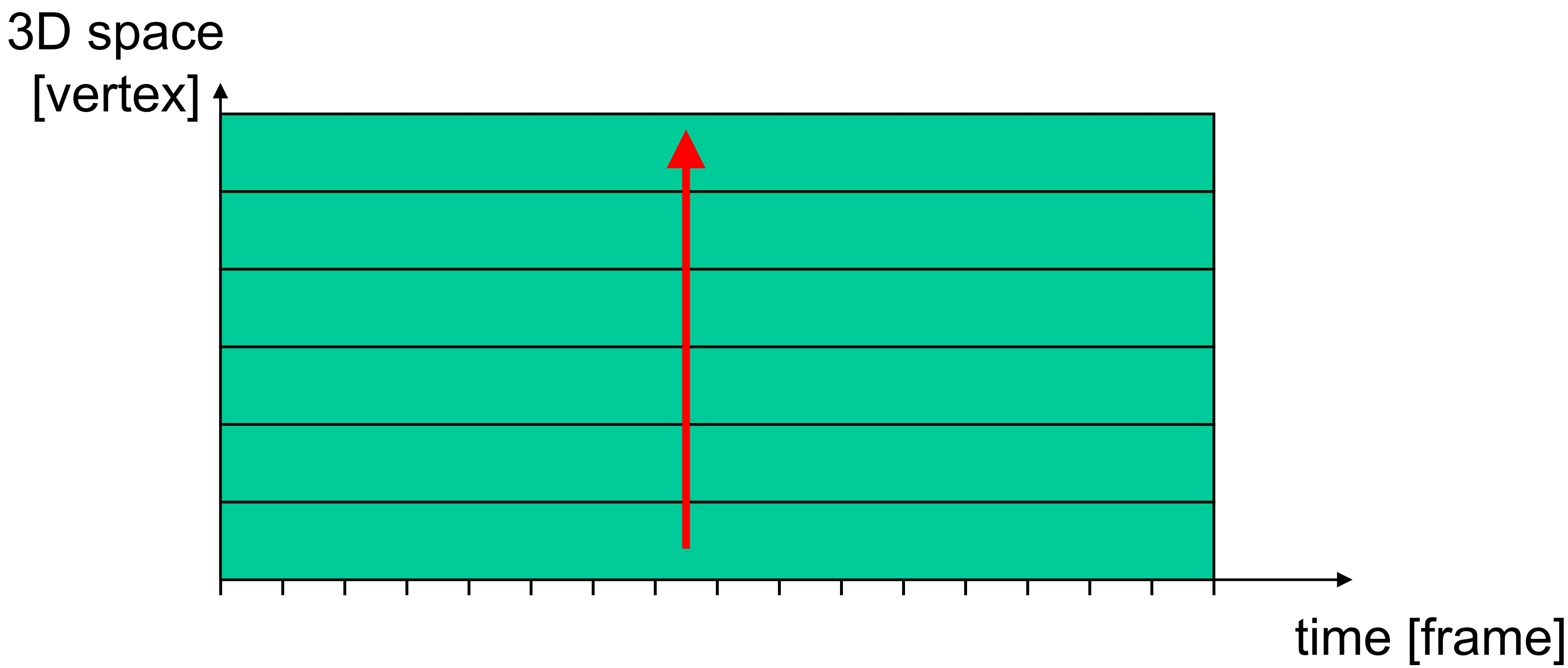


Scalable Compression Types

Spatial scalability on animation level

– successive decoding increases the number of vertices of the entire animation

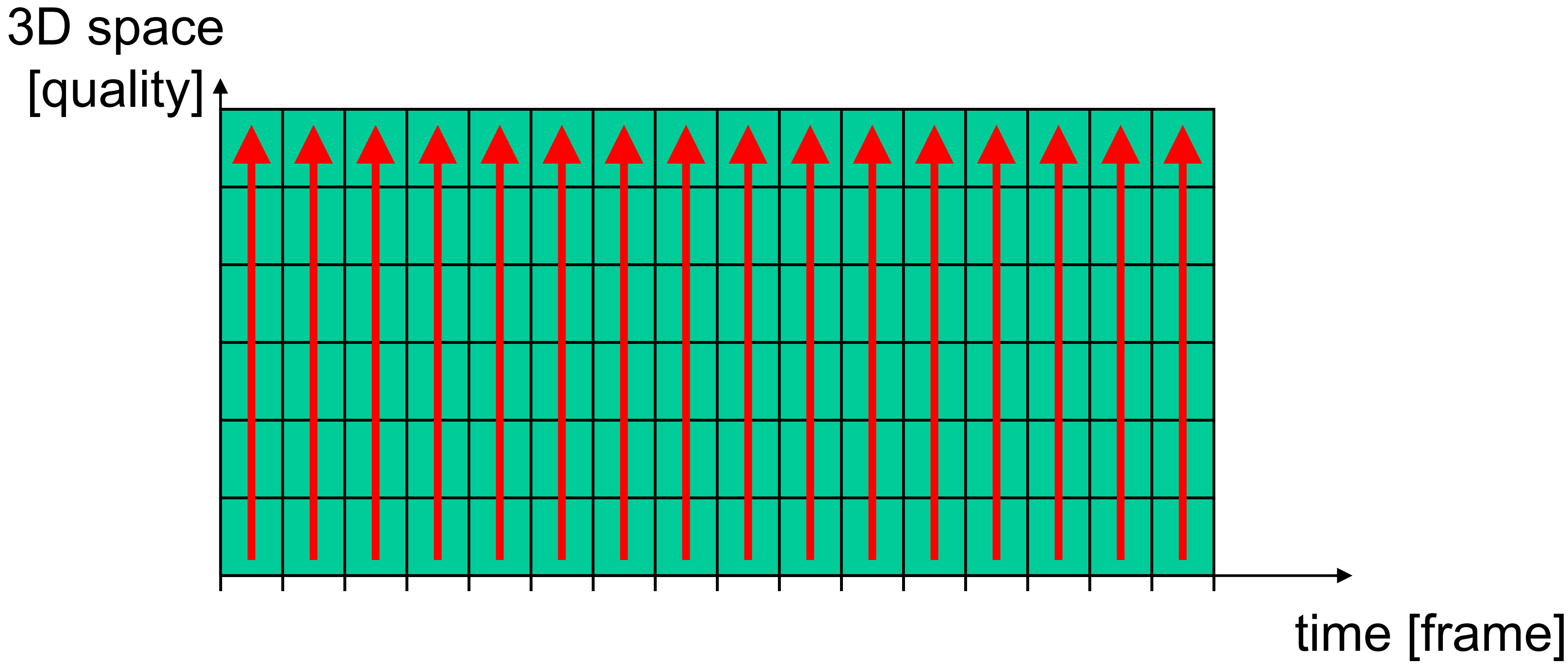
– no frame-wise decoding possible



Scalable Compression Types

Quality scalability on frame level

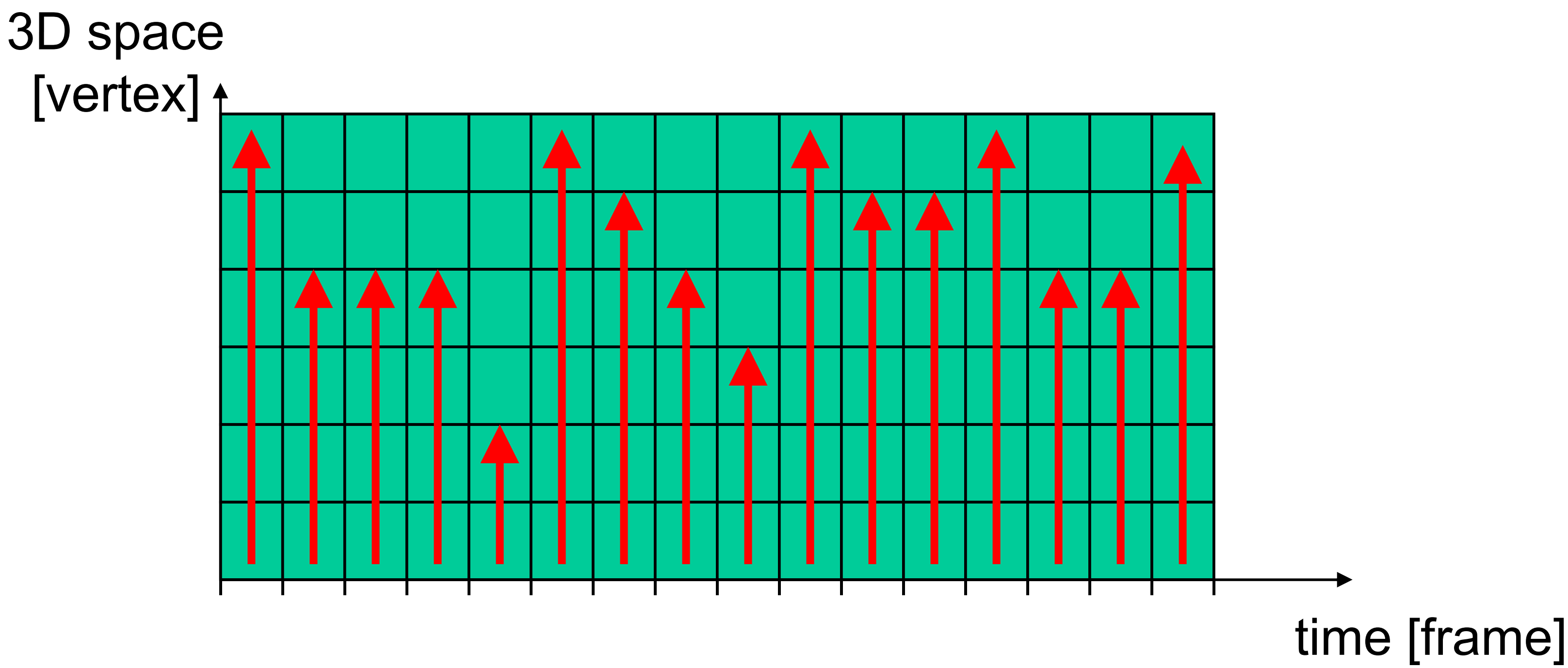
- frame-wise decoding
- successive decoding increases the accuracy of vertex positions per frame

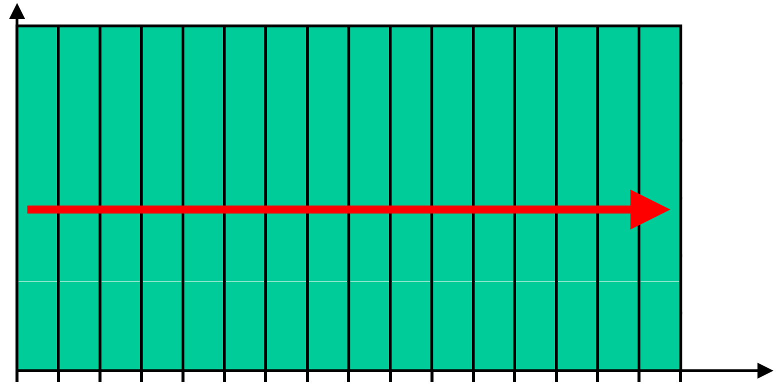


Scalable Compression Types

Spatial scalability on frame level

- frame-wise decoding
- successive decoding increases the number of vertices per frame



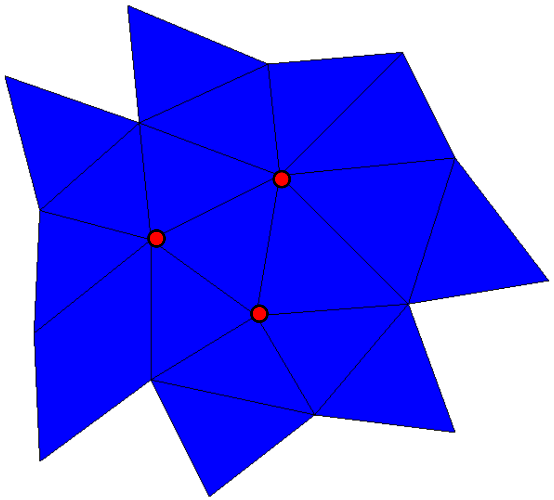


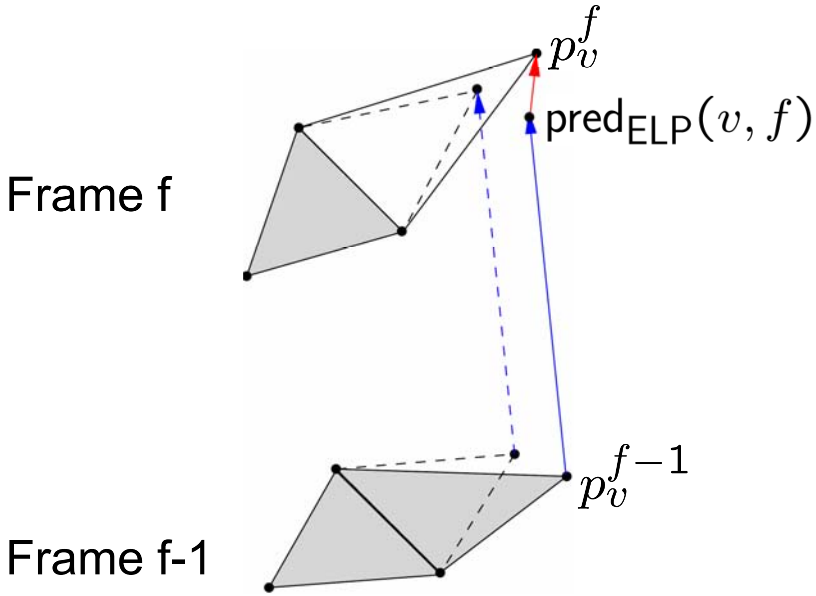
Single-Rate Compression Approaches

Dynapack

- **Dynapack** [Ibarria & Rossignac ‘03]

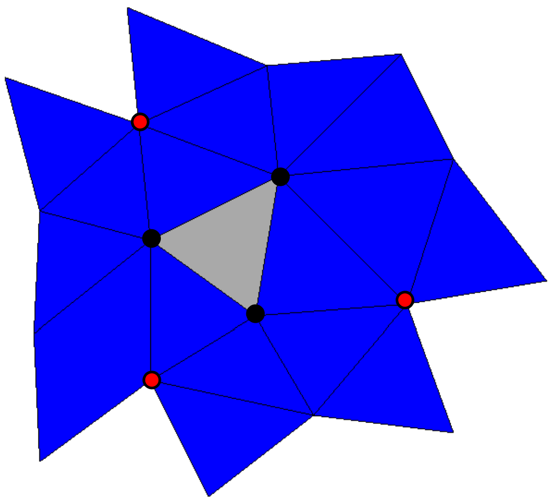
- Traverse all vertices in each frame in a deterministic order
  – use Edgebreaker-like vertex traversal
  – order of vertex traversal defines order of compression of vertex positions within a frame
  – predict next vertex using already encoded vertex positions of current and previous frames
  – encode prediction error

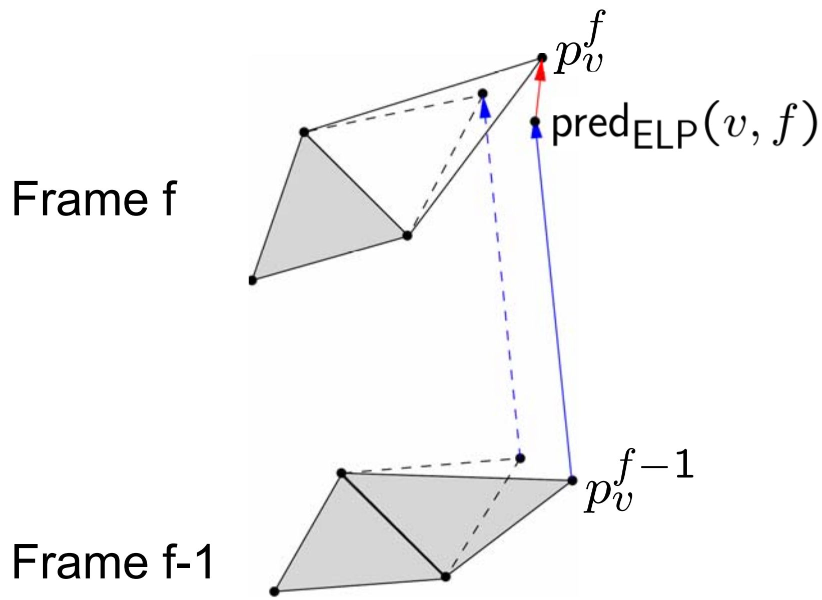




Dynapack

- **Dynapack** [Ibarria & Rossignac ‘03]
- Traverse all vertices in each frame in a deterministic order
  - use Edgebreaker-like vertex traversal
  - order of vertex traversal defines order of compression of vertex positions within a frame
  - predict next vertex using already encoded vertex positions of current and previous frames
  - encode prediction error

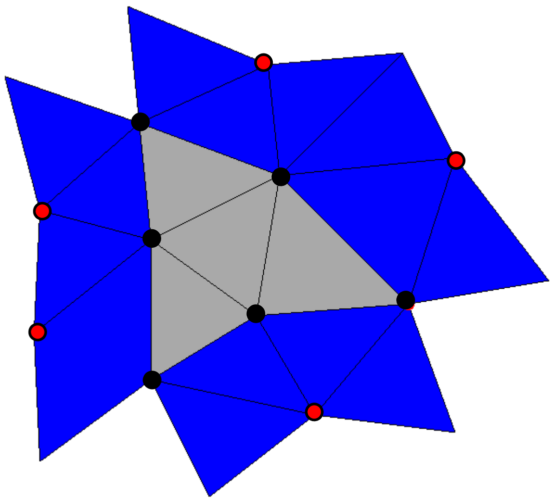


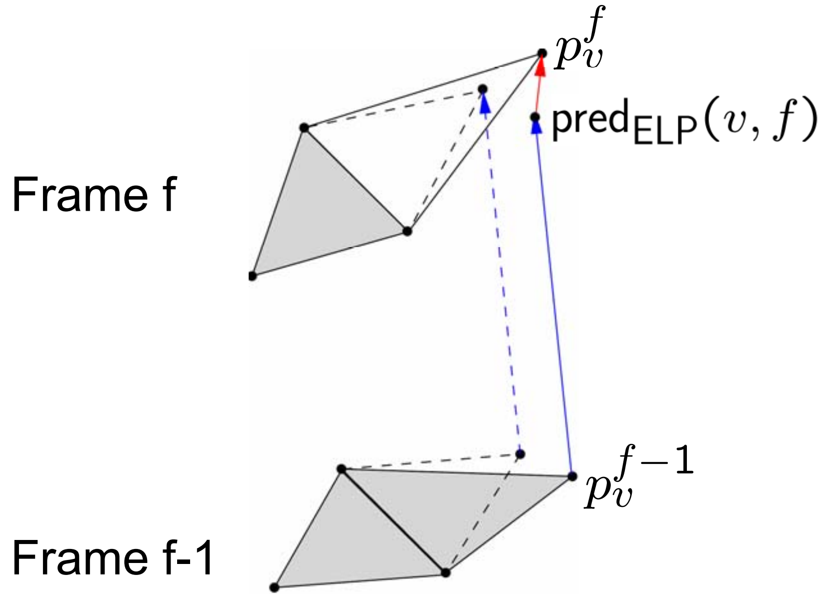


Dynapack

- **Dynapack** [Ibarria & Rossignac '03]

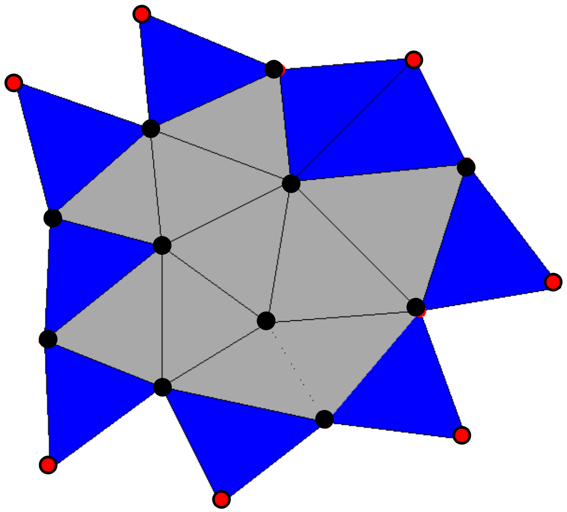
- Traverse all vertices in each frame in a deterministic order
  - use Edgebreaker-like vertex traversal
  - order of vertex traversal defines order of compression of vertex positions within a frame
  - predict next vertex using already encoded vertex positions of current and previous frames
  - encode prediction error

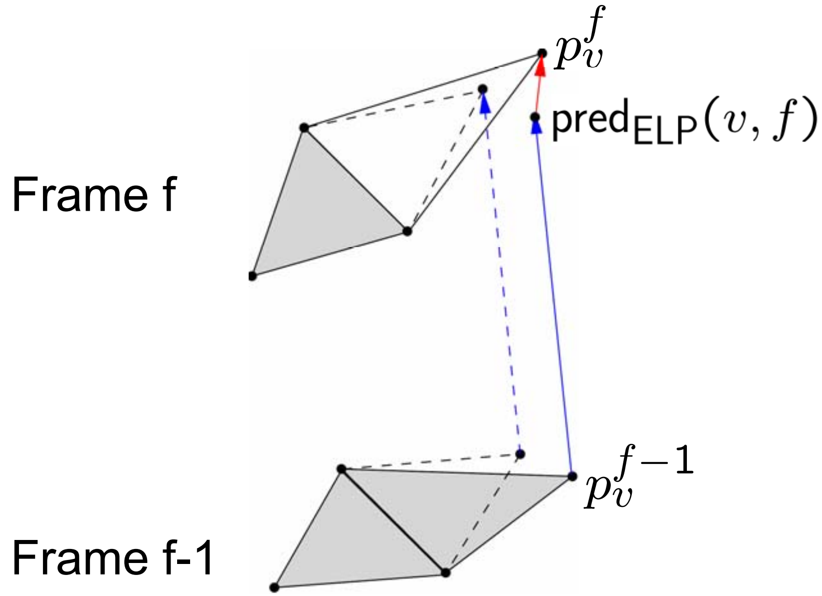




Dynapack

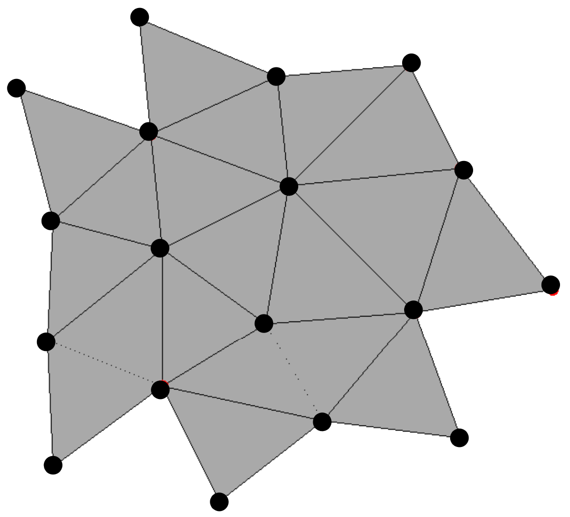
- **Dynapack** [Ibarria & Rossignac ‘03]
- Traverse all vertices in each frame in a deterministic order
  - use Edgebreaker-like vertex traversal
  - order of vertex traversal defines order of compression of vertex positions within a frame
  - predict next vertex using already encoded vertex positions of current and previous frames
  - encode prediction error

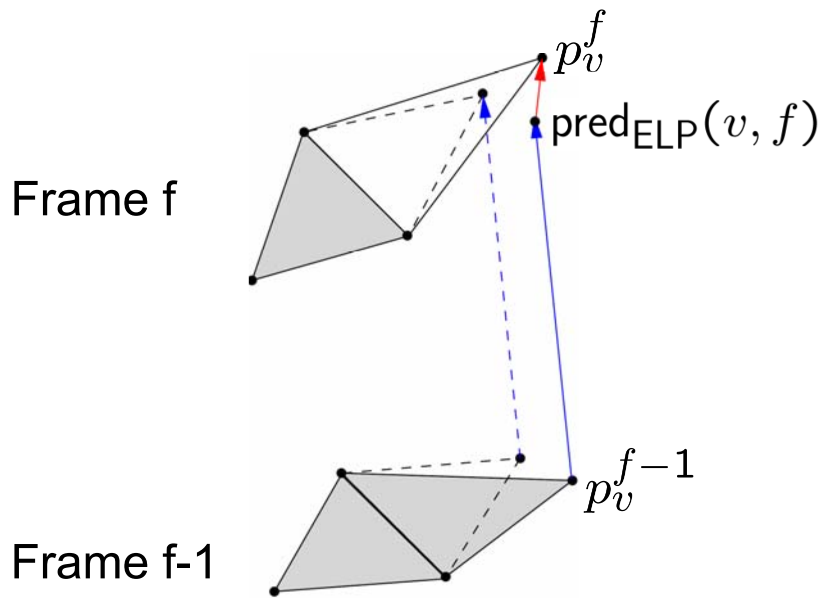




Dynapack

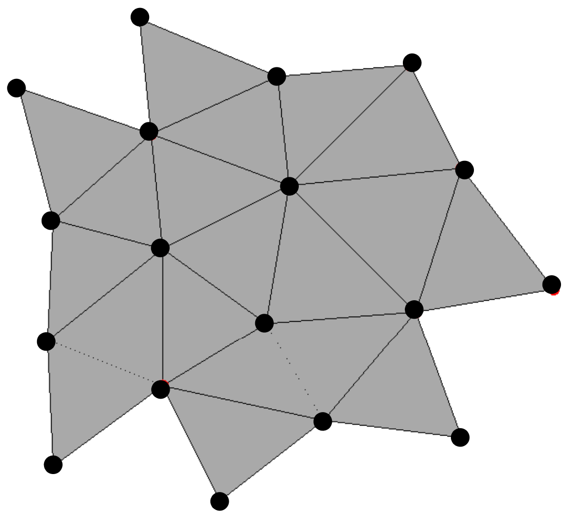
- **Dynapack** [Ibarria & Rossignac ‘03]
- Traverse all vertices in each frame in a deterministic order
  - use Edgebreaker-like vertex traversal
  - order of vertex traversal defines order of compression of vertex positions within a frame
  - predict next vertex using already encoded vertex positions of current and previous frames
  - encode prediction error

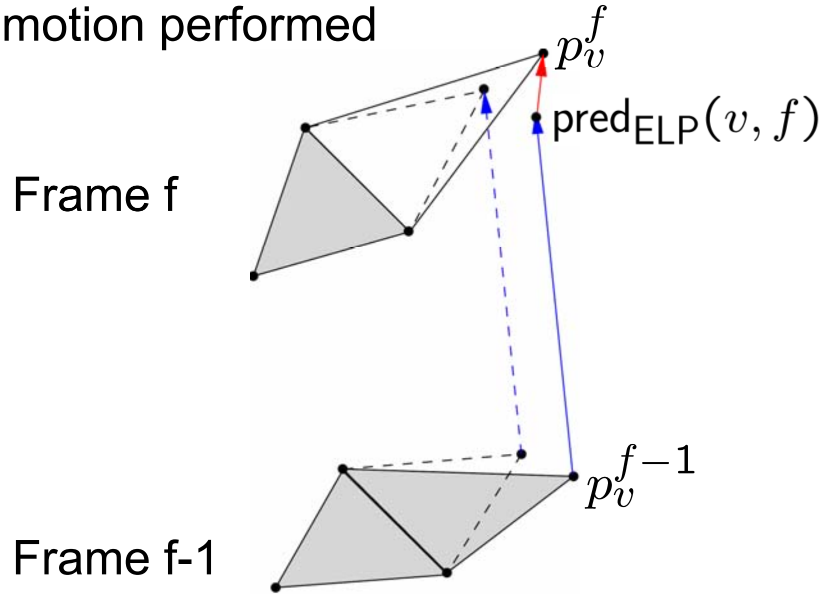




Dynapack

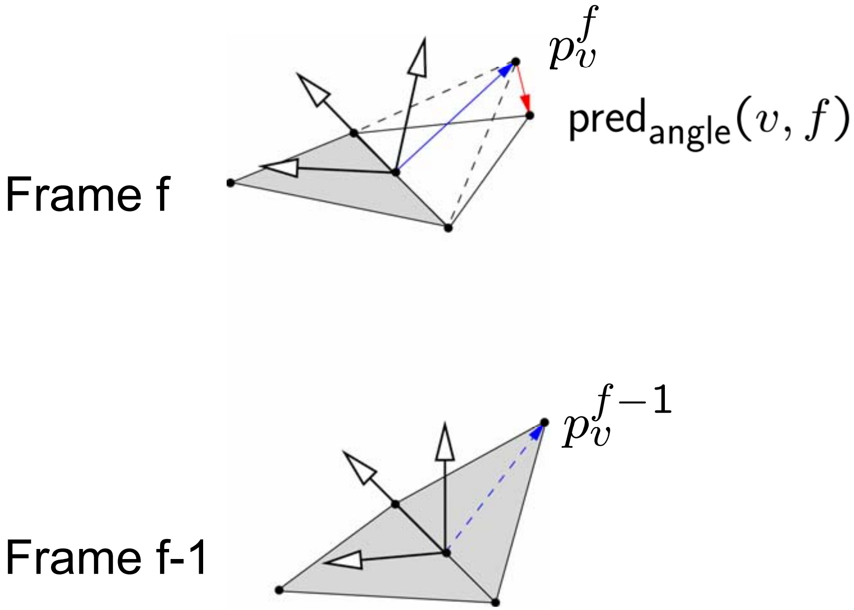
- Encoding of vertex positions:
  - pre-quantize all vertex positions in dynamic mesh
  - losslessly encode quantized vertex locations
  - use linear spatio-temporal predictor: *Extended Lorenzo Predictor*
    - extension of the parallelogram-predictor (pure spatial predictor)
  - entropy encode prediction error
  - error free prediction if translational motion performed





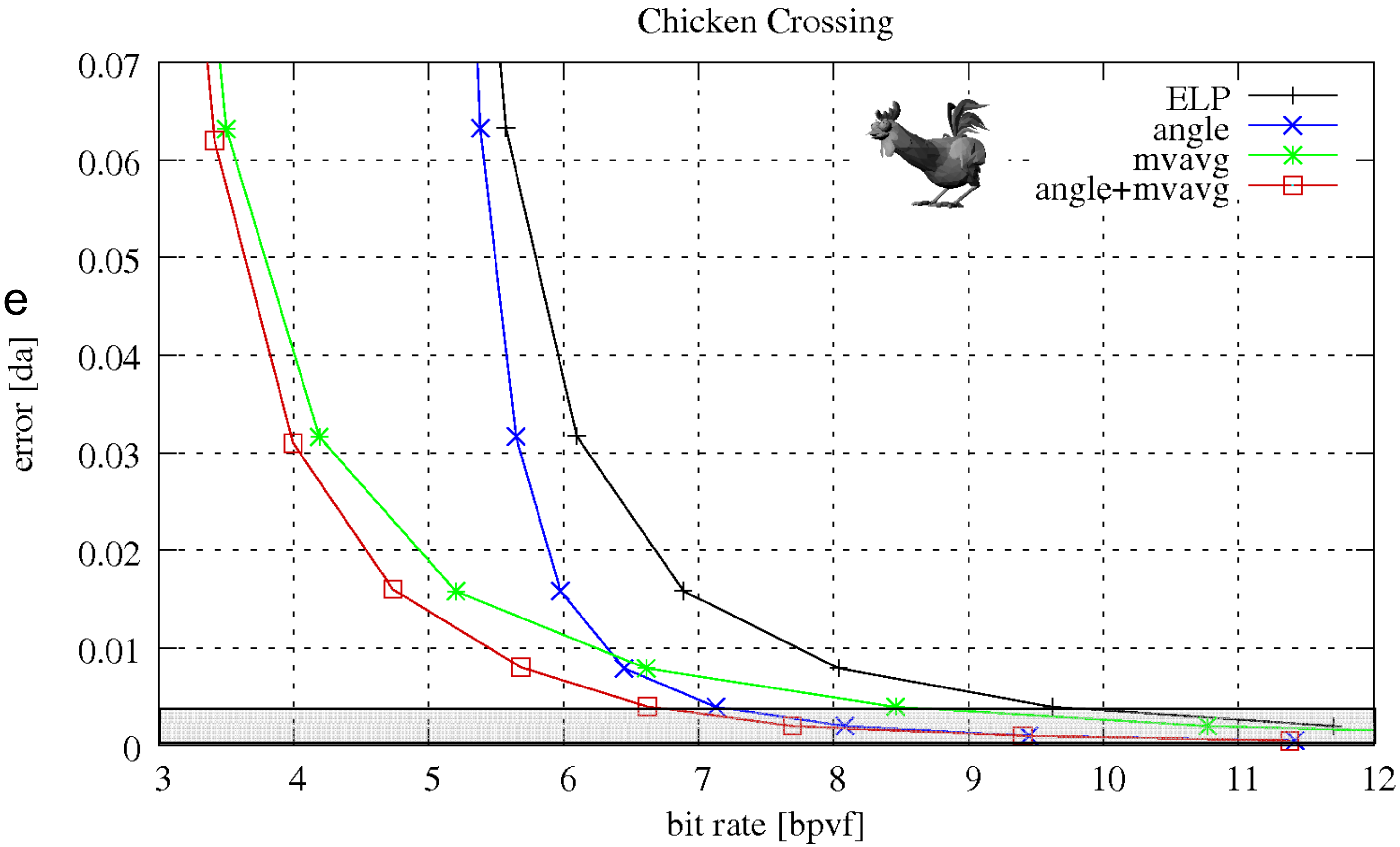
Angle Preserving Predictor

- **Angle Preserving Predictor** [Stefanoski & Ostermann '06]
  - preserve dihedral angle between triangles in frame f-1 in frame f when predicting
  - use local coordinate frames to calculate predicted value
  - entropy encode prediction error
  - error free prediction if performing rigid body motion



Angle Preserving Predictor

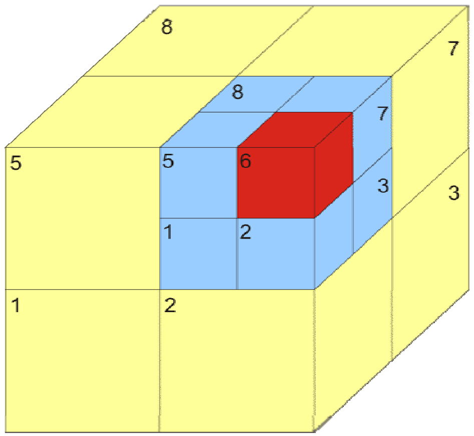
- Error measure: vertex-wise L_2
- Bit rate: bit per vertex and frame [bpvf]

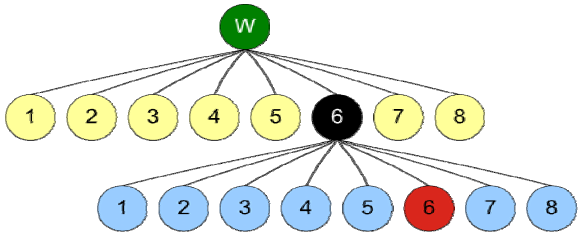


- significant compression gains of the *Angle Preserving Predictor* against the *Extended-Lorenzo Predictor* in the domain of bit-rates over 7 bpvf

Differential 3D Mesh Coding

- **D3DMC** [Müller et. al ´04]
  - exploitation of temporal and spatial statistical dependencies between adjacent frames
  - description of the motion of all vertices of a frame using an octree
    - vertex positions and corresponding motion vectors of a frame are stored in the octree
    - vertices inside of a subcube perform homogeneous motion





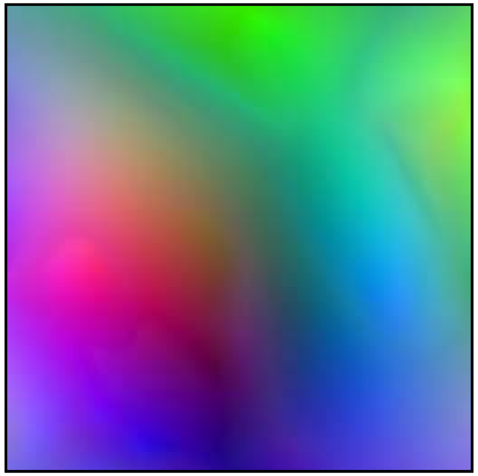
Differential 3D Mesh Coding

- **Introduction of different prediction modes** for each subcube
  - **Intra** : The coordinates of all vertices in the current subcube are stored
  - **Delta** : The difference vectors of all vertices in the current subcube between the previous and the current mesh are stored
  - **Linear** : Usage of a single estimated motion vector to describe the movement of all vertices in the current subcube
  - **Tri-linear** : Usage of eight estimated motion vectors to describe the movement of all vertices in the current subcube
- **Linear quantization and entropy coding** of the estimated motion vectors. Usage of CABAC for entropy coding.

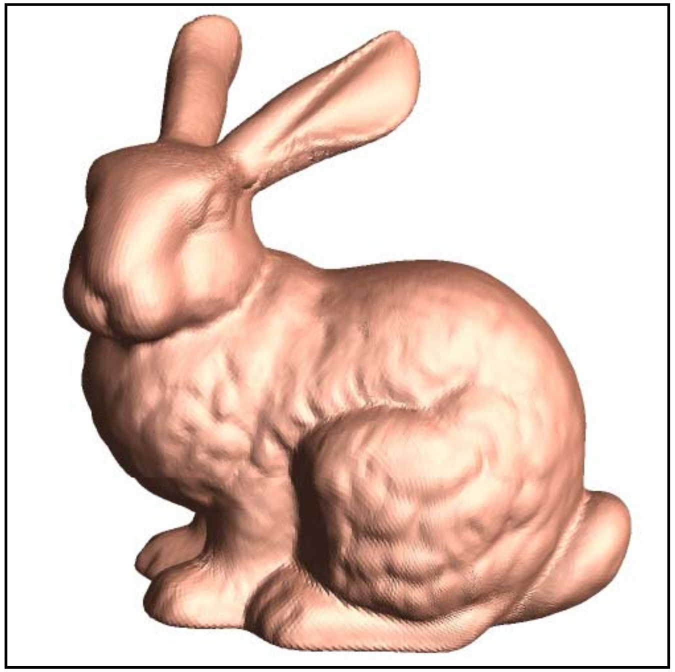
- RD optimized version of D3DMC increases compression efficiency compared to fixed D3DMC
  - generic: no exploitation of mesh connectivity
  - computationally exhaustive calculations necessary to determine optimized octree structure

Geometry Videos

- **Geometry Videos** [Briceno et. al ‘03]
  - based on geometry images
  - basic idea: interpret (R,G,B) values in image as (X,Y,Z) coordinate in 3D space
  - a dynamic mesh is represented as sequence of geometry images

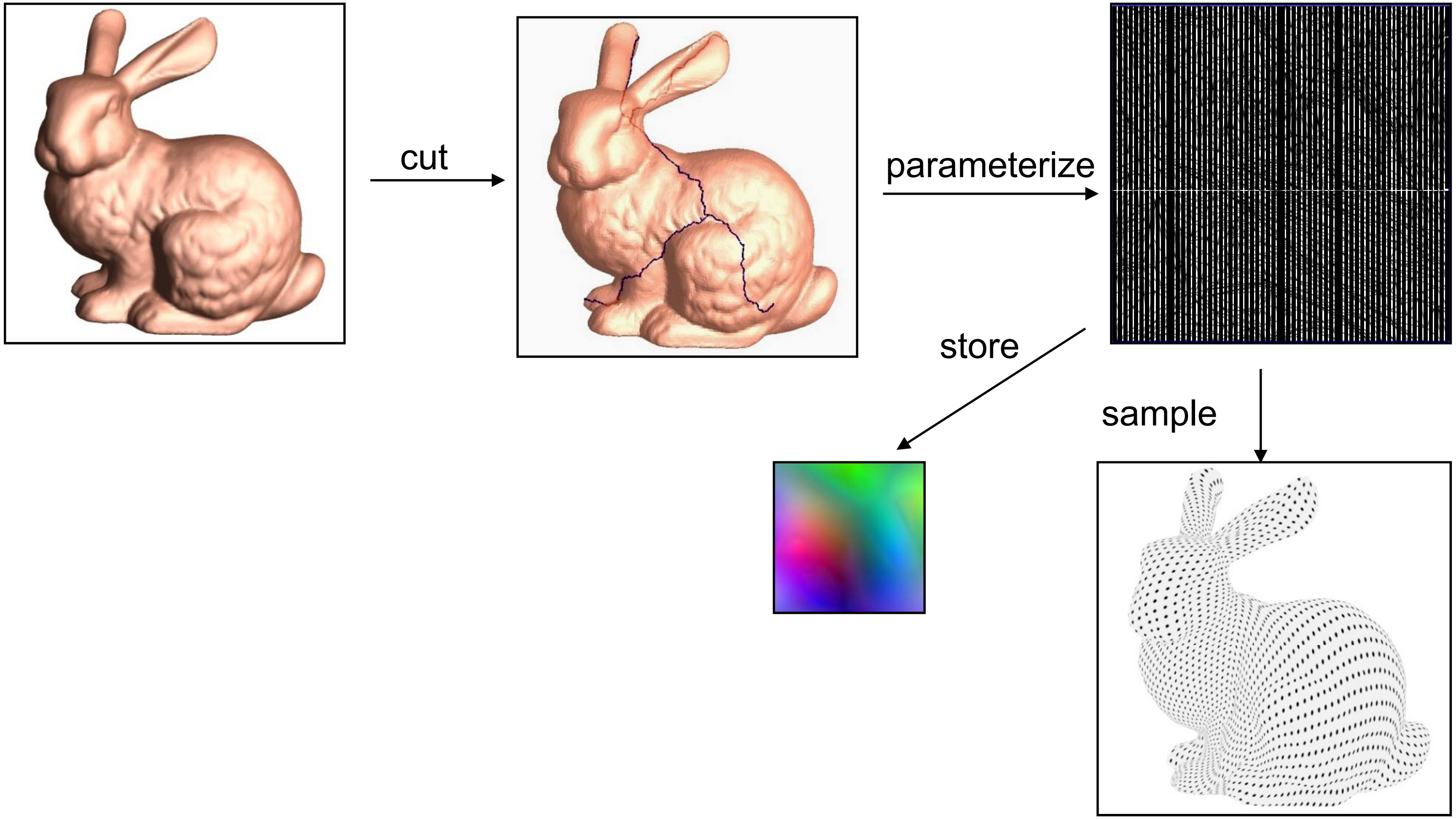


(R,G,B) to (X,Y,Z)



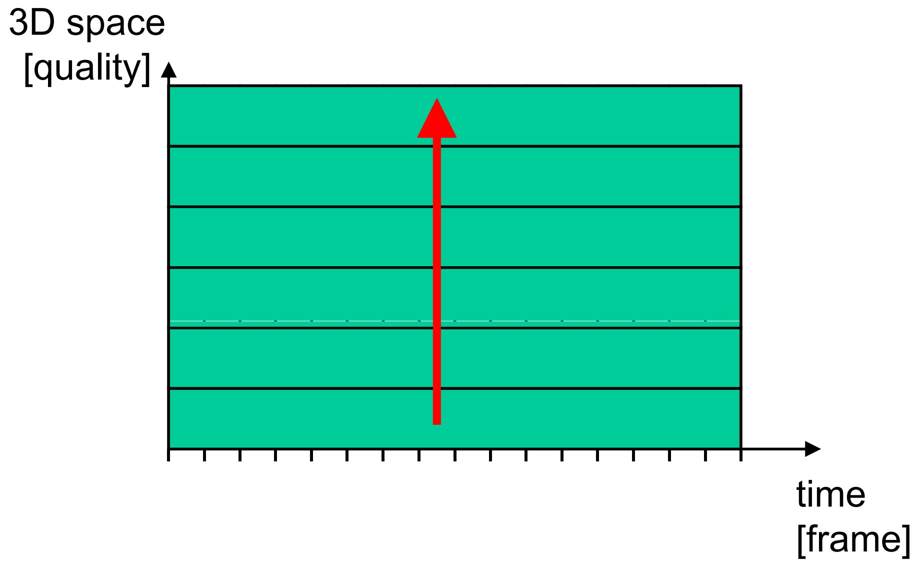
Geometry Videos

- How to get a geometry image representation?



Geometry Videos

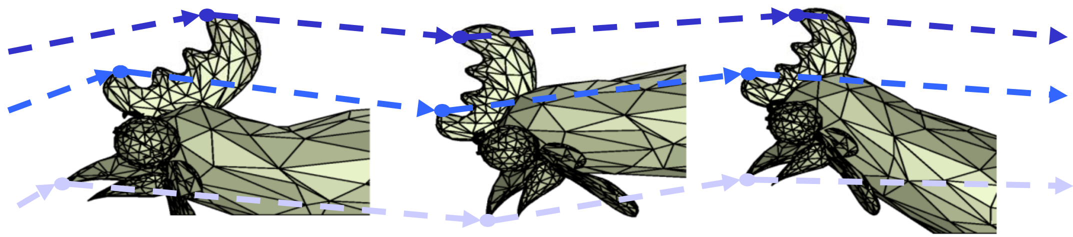
- This process is repeated for all frames in a dynamic mesh in order to create a sequence of geometry images (2D video)
- Compression of sequence of geometry images
  - use existent video coding algorithms (e.g. MPEG-4)
- Pros
  - fast decoding
  - use of standardized off-the-shelf video coders/decoders
- Cons
  - temporally consistent parameterization required for efficient compression at encoder side
    - calculation of parameterization is computationally very demanding
    - artifacts due to topological incompatibilities between dynamic mesh and parameter domain (conceptual problem)



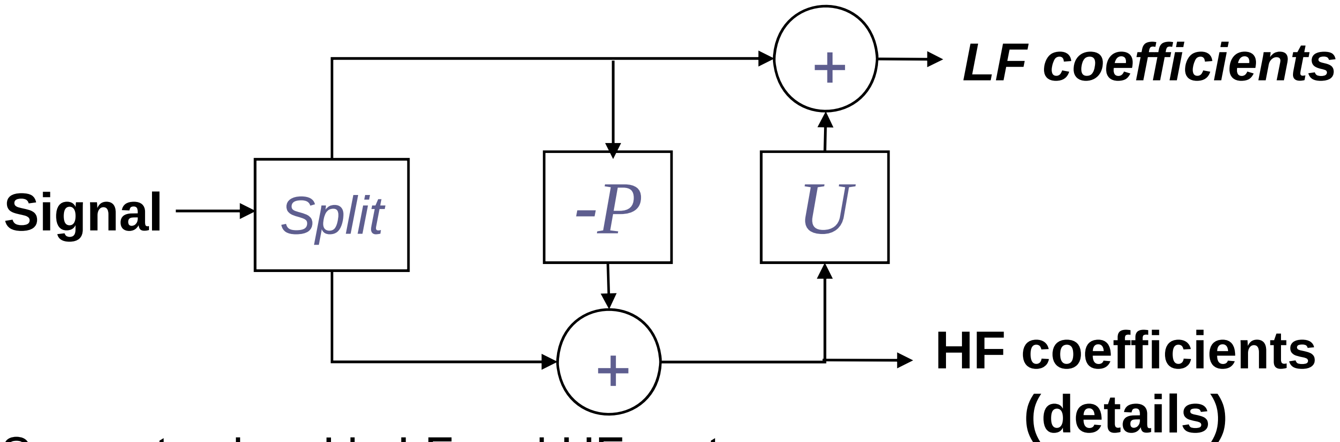
Quality Scalable Approaches on Animation Level

Temporal Wavelet Compression (TWC)

- **TWC** [Payan & Antonini ’05]



- encode 1D signals (3V different signals in temporal direction)
- Traditional 1D lifting scheme in temporal direction



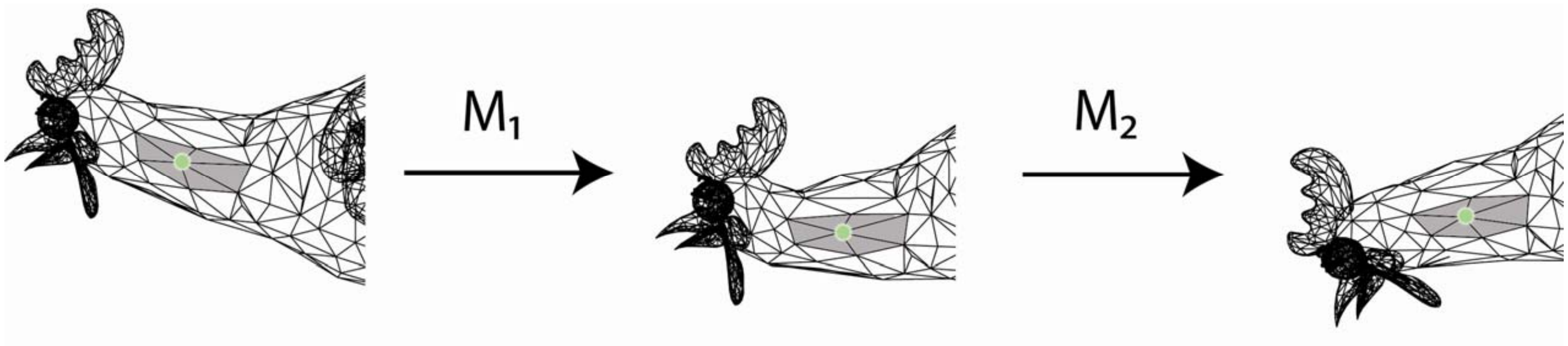
- Separate signal in LF and HF part
- Cascade lifting scheme at LF part → $HF_{L-1}, \ldots, HF_2, HF_1, HF_0, LF_0$

Temporal Wavelet Compression (TWC)

- employ scalar quantization
  - find quantization steps for LF_0 and $HF_0,\ldots,HF_{L-1}$ that optimize MSE of reconstructed dynamic mesh
    - constraint: user given bit-budget
  - solution: Lagrangian optimization
- encode successively all $LF_0,HF_0,\ldots,HF_{L-1}$ coefficients for all trajectories
- approach allows quality scalability on animation level
  - the more HF_i coefficients are decoded the higher the accuracy of all vertex positions
- approach allows temporal scalability on animation level
  - increase frame rate while decoding more HF_i coefficients

Skinning-based compression

- **Skinning** [Mamou et al. ‘06]
- motion-based segmentation
  - each vertex assigned a sequence of matrices representing the movement of its local neighborhood
  - k-means clustering applied to the set of matrix sequences



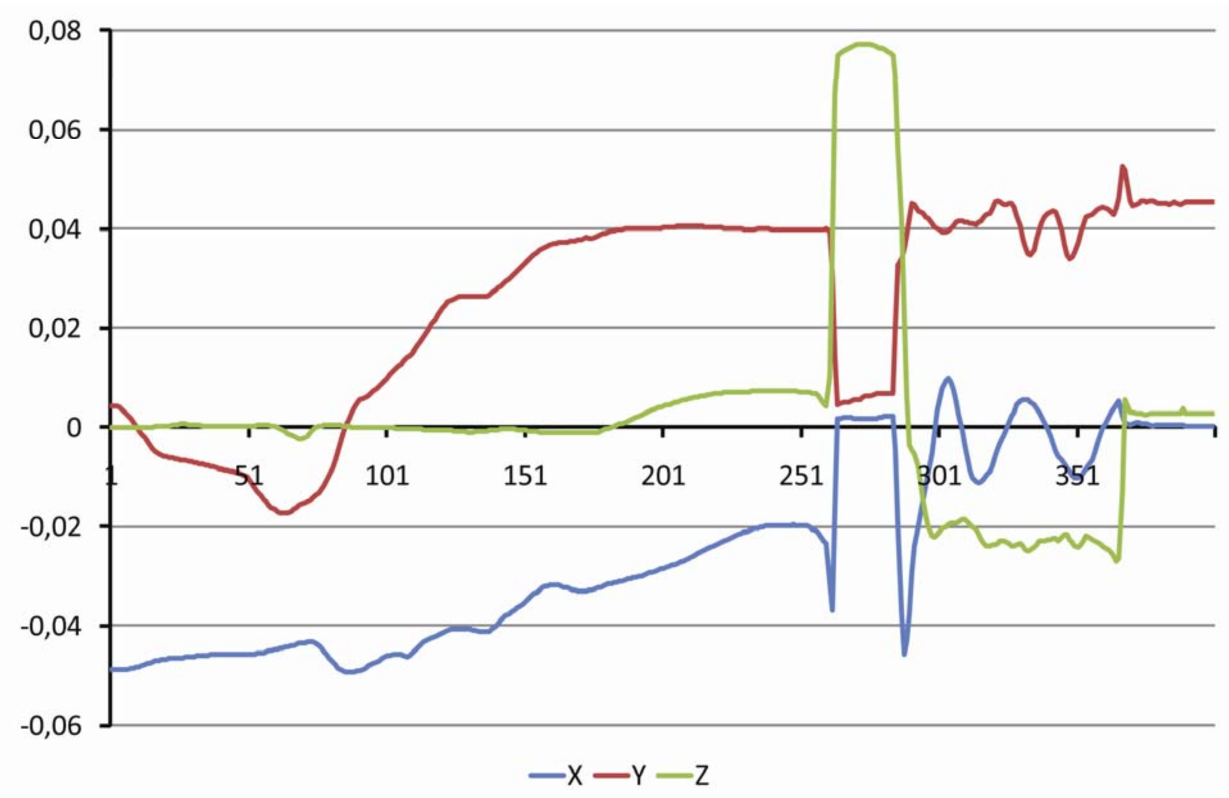
=> set of segments (clusters)

Skinning-based compression

- affine motion estimation
  - each segment assigned a single sequence of matrices which best describes its movement
  - least squares fitting
- skinning model
  - each vertex assigned a set of weights
  - only neighboring segments (low number)
  - skinning coefficients constant for the sequence
- prediction residuals
  - entropy coded

Trajectory space PCA

- sample = trajectory of a single vertex
- EigenTrajectories



Trajectory space PCA

- usually more samples than dimensions
- EVD for PCA
- lower computational complexity
  – autocorrelation matrix of size 3Fx3F
- problems:
  – cannot be extracted frame-by-frame (longer sequences cut)
  – incomplete exploitation of spatial coherence

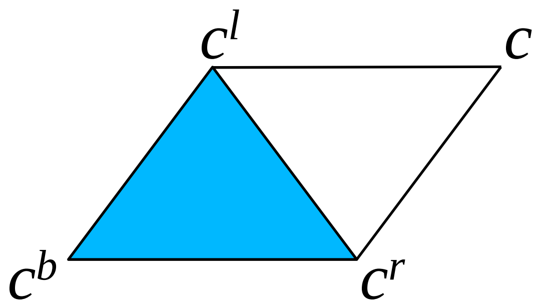
Clustered PCA approach

- **Clustering & EigenTrajectories** [Sattler et al. '05]
- trajectory-space PCA
- added clustering
  - separate PCA in each cluster
- partial exploitation of spatial coherence
  - vertices in the same cluster have a specific PCA
- problems with clustering
  - random seeds lead to significantly different clusterings

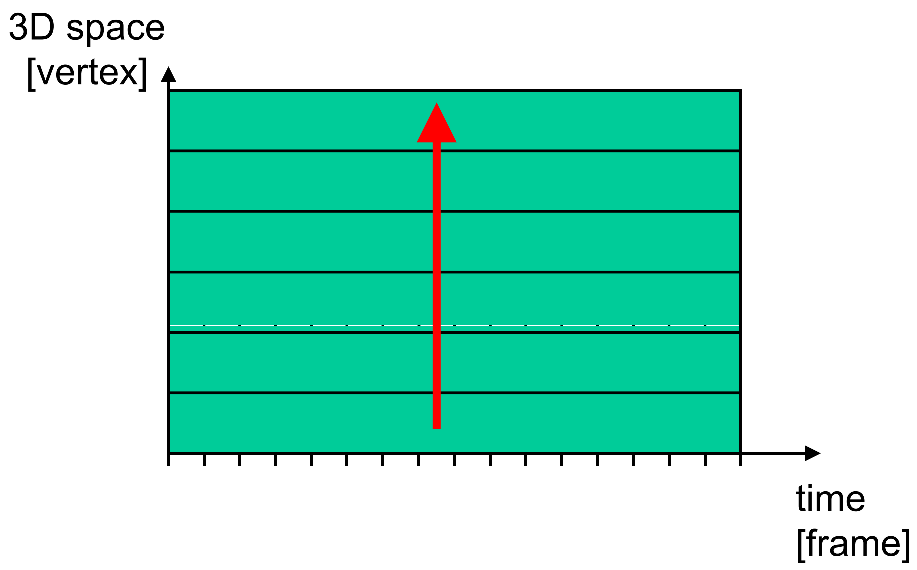
Coddyac

- **Extrapolation & EigenTrajectories** [Vasa & Skala '07]
- trajectory space PCA
- coefficients for each vertex are predicted from neighbours during EdgeBreaker pass through the mesh

$$c_i = c_i^l + c_i^r - c_i^b$$



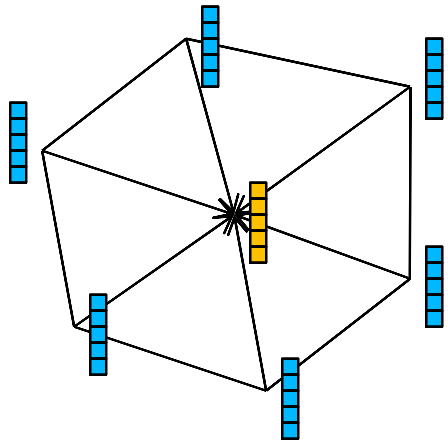
- exploits spatial coherence
- no need for clustering
- lower residual entropy
  - proportional to edge length, rather than cluster radius



Spatially Scalable Approaches on Animation Level

Combined compression and simplification (CoCoS ☺)

- **Decimation + PCA** [Vasa & Skala '08]
- spatially scalable
- extrapolation replaced by interpolation

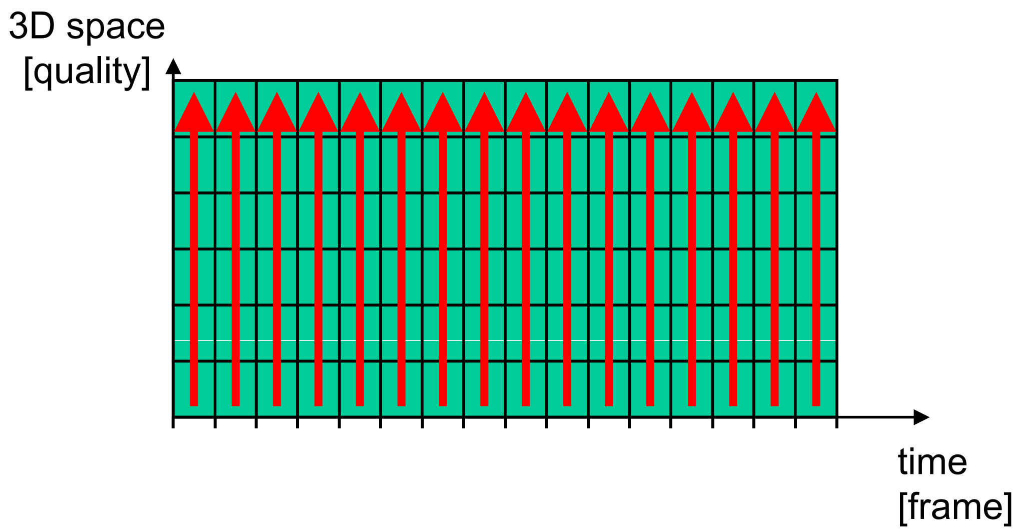


- algorithm:
  1. transmit full connectivity
  2. decimate connectivity
  3. transmit PCA coefs for coarse version using paralellogram prediction (=Coddyac)
  4. inverse decimation, using neighborhood average

Combined compression and simplification (CoCoS ☺)

- decimation step
  - connectivity driven (keep degree close to 6)
  - partially driven by the encoder
    - decimation level transmitted
    - preserve correct geometry
    - avoid sharp triangles
    - simplification order driven by prediction accuracy
    - different simplification strategies can be used

- prediction step
  - neighborhood average applied on PCA coefficients
  - interpolation more robust than extrapolation

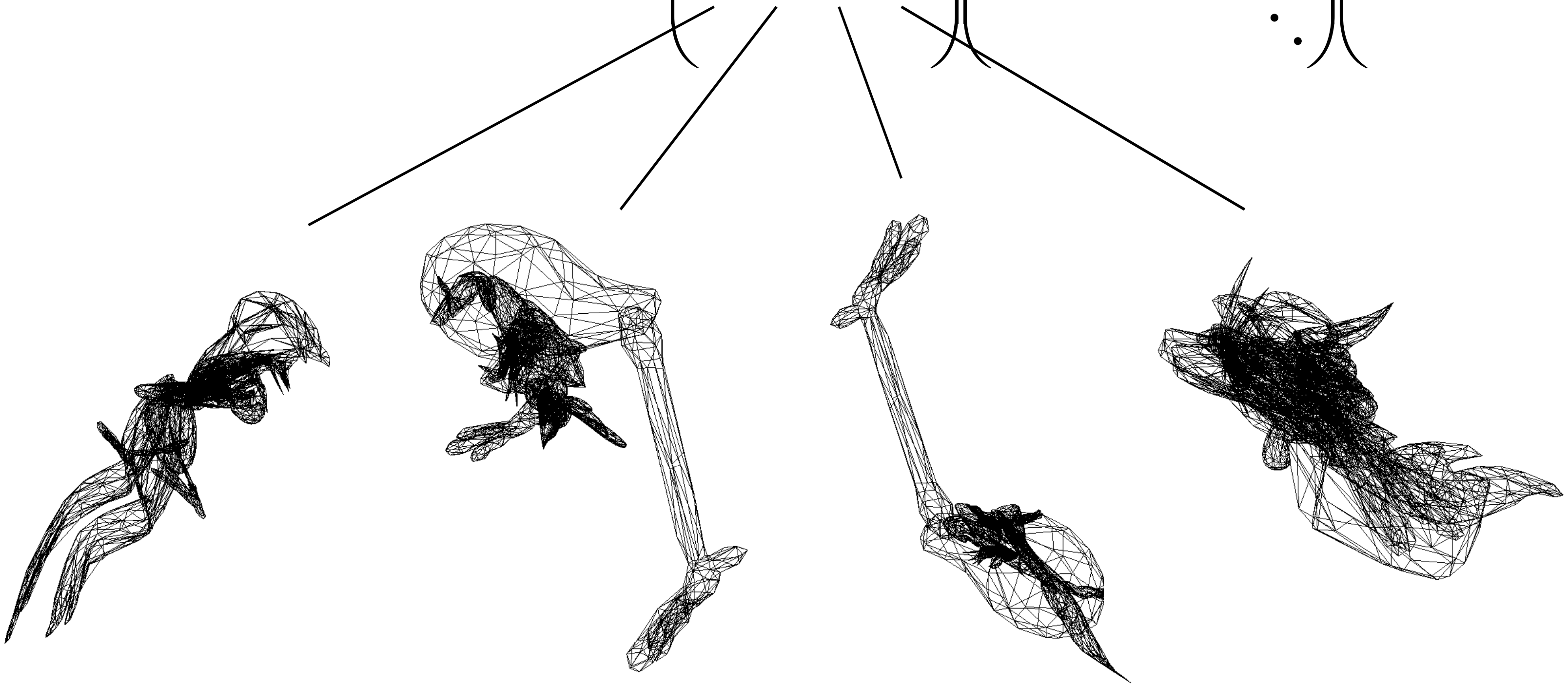


Quality Scalable Approaches on Frame Level

Shape space PCA

- sample = single frame of animation
- EigenShapes

$$B = \begin{pmatrix} | & | & | \end{pmatrix} \begin{pmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \; \end{pmatrix}$$



Shape space PCA

- **Dimensionality reduction & EigenShapes**
  [Alexa & Müller '00]
- rigid motion compensation
  - least suqres fitting
  - affine transform for each frame
- PCA
  - usually more dimensions than samples
  - SVD for PCA
- high computational complexity
- temporal coherence not exploited

Shape-space PCA + LPC

- **LPC & EigenShapes** [Karni & Gotsman '04]
- exploits temporal coherence of the PCA coefficients
- apply Linear Predictive Coding on the PCA coefficients
  - optimized predictor for dependent sequences of values

f1	f2	f3	...
X	X	X	...
Y	Y	Y	...
Z	Z	Z	...
X	X	X	...
Y	Y	Y	...
Z	Z	Z	...
...	...	...	...

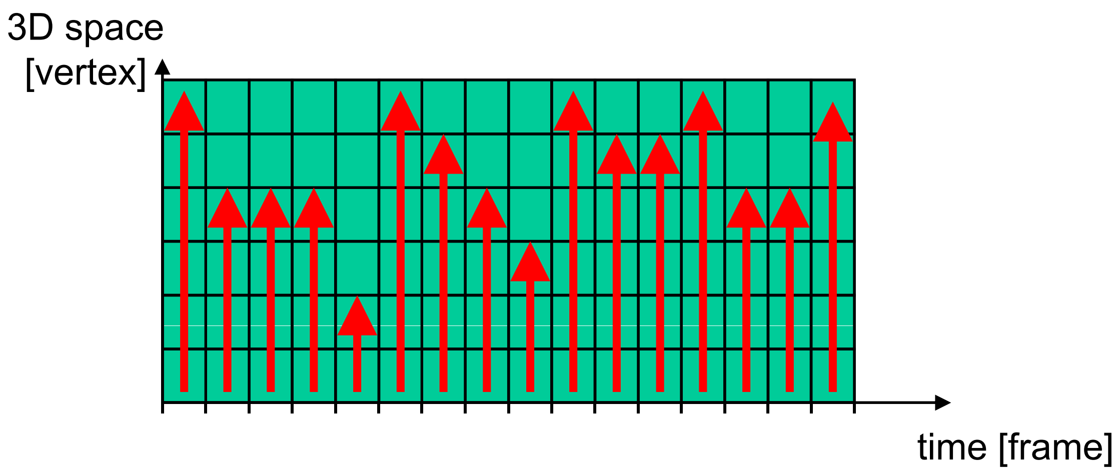
$\xrightarrow{\text{PCA}}$

f1	f2	f3	...
C1	C1	C1	...
C2	C2	C2	...
C3	C3	C3	...
...	...	...	...

$\xrightarrow{\text{LPC}}$

f1	f2	f3	...
C1	ΔC1	ΔC1	...
C2	ΔC2	ΔC2	...
C3	ΔC3	ΔC3	...
...	...	...	...

- problem with large size of PCA basis vectors



Spatially Scalable Approaches on Frame Level

Spatial Wavelet Compression

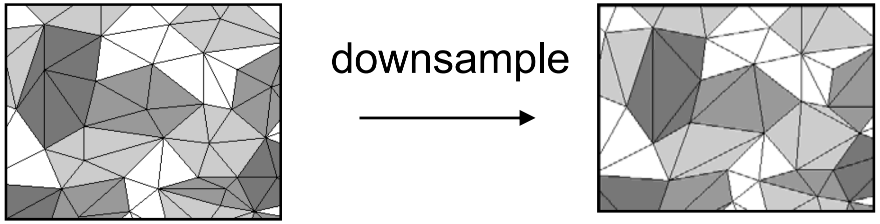
- **AWC** [Guskov & Khodakovsky '04]
- encode first frame's connectivity and geometry using a static mesh coder
- derive spatial predictors (for all frames) using first frame's geometry
  - downsample first frame using a sequence of mesh simplifications
  - calculate prediction coefficients (minimize discrete fairing functional)

- encode remaining frames using an open DPCM loop (Wavelet Transform)
  - spatial decorrelation
  - encoding order = inverse order of downsampling
- encode remaining residuals using differential coding in temporal direction
  - temporal decorrelation

Layered Dynamic 3D Mesh Coding (LD3DMC)

- **LD3DMC** (Stefanoski et. al '07)

- patch based decomposition and connectivity based simplification used for downsampling (def. spatial layers)



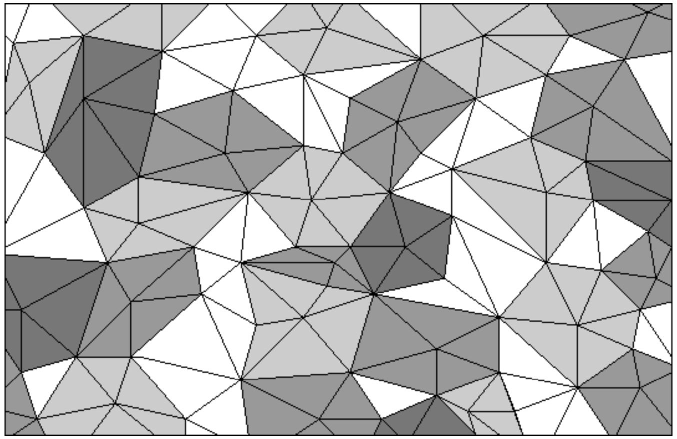
- closed DPCM loop to exploit spatio-temporal dependencies between frames and spatial layers
  - usage of local spatio-temporal predictors
    - linear and non-linear

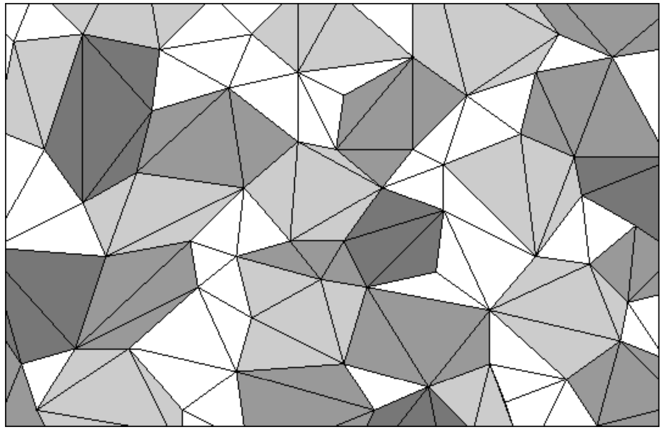
- the LD3DMC coding structure is part of MPEG 4, Part 16 (AFX) referred to as FAMC.

Layered Dynamic 3D Mesh Coding (LD3DMC)

Patch based decomposition and vertex removal:

K_l $\xrightarrow{V_l}$ K_{l-1}

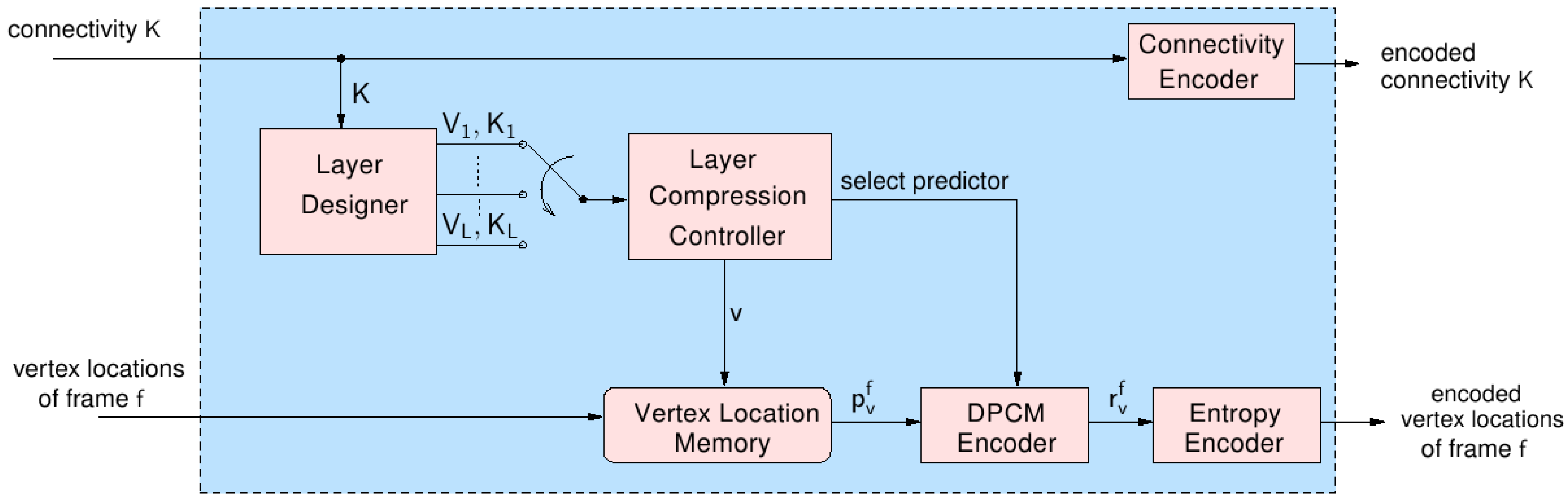




- removed vertices compose layer V_l
- an iteration of this simplification procedure results in a disjoint decomposition of $V=\cup V_l$
- key property of patch based simplification:

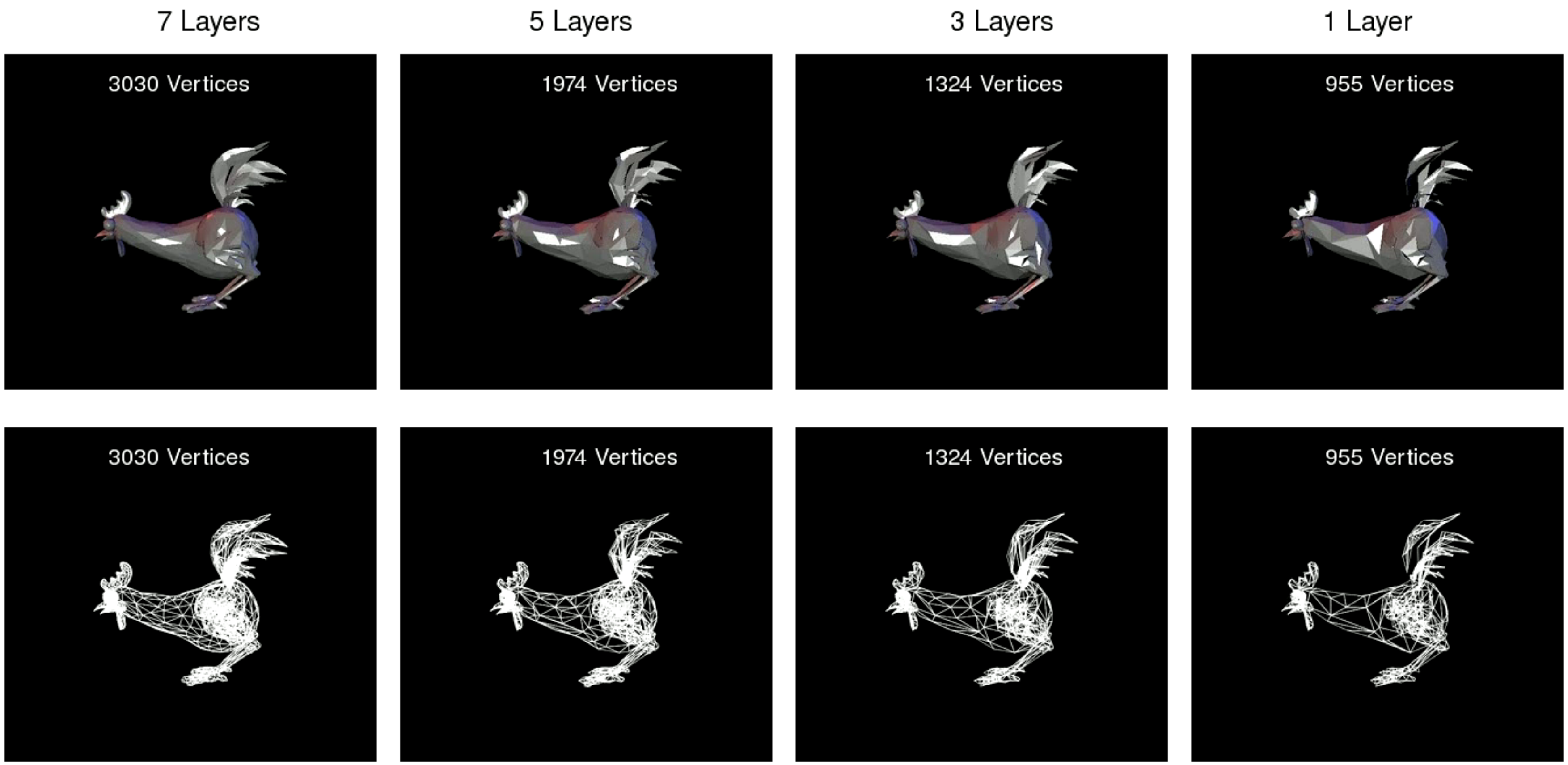
> all neighbors of a vertex v in V_l are contained in K_{l-1}

Layered Dynamic 3D Mesh Coding (LD3DMC)



V_1	$V_1 \cup V_2$	$V_1 \cup V_2 \cup V_3$
K_1	K_2	K_3

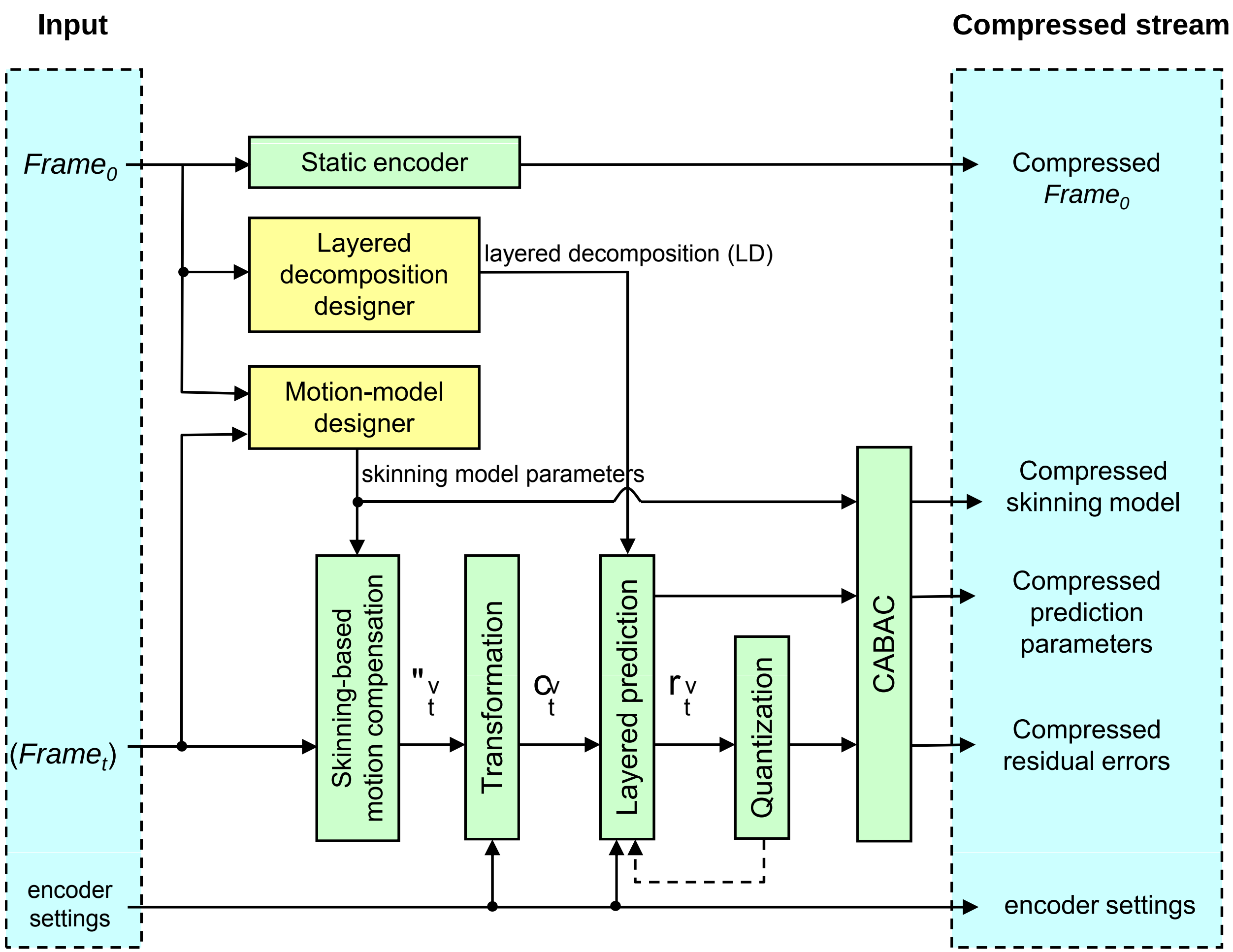
Layered Dynamic 3D Mesh Coding (LD3DMC)



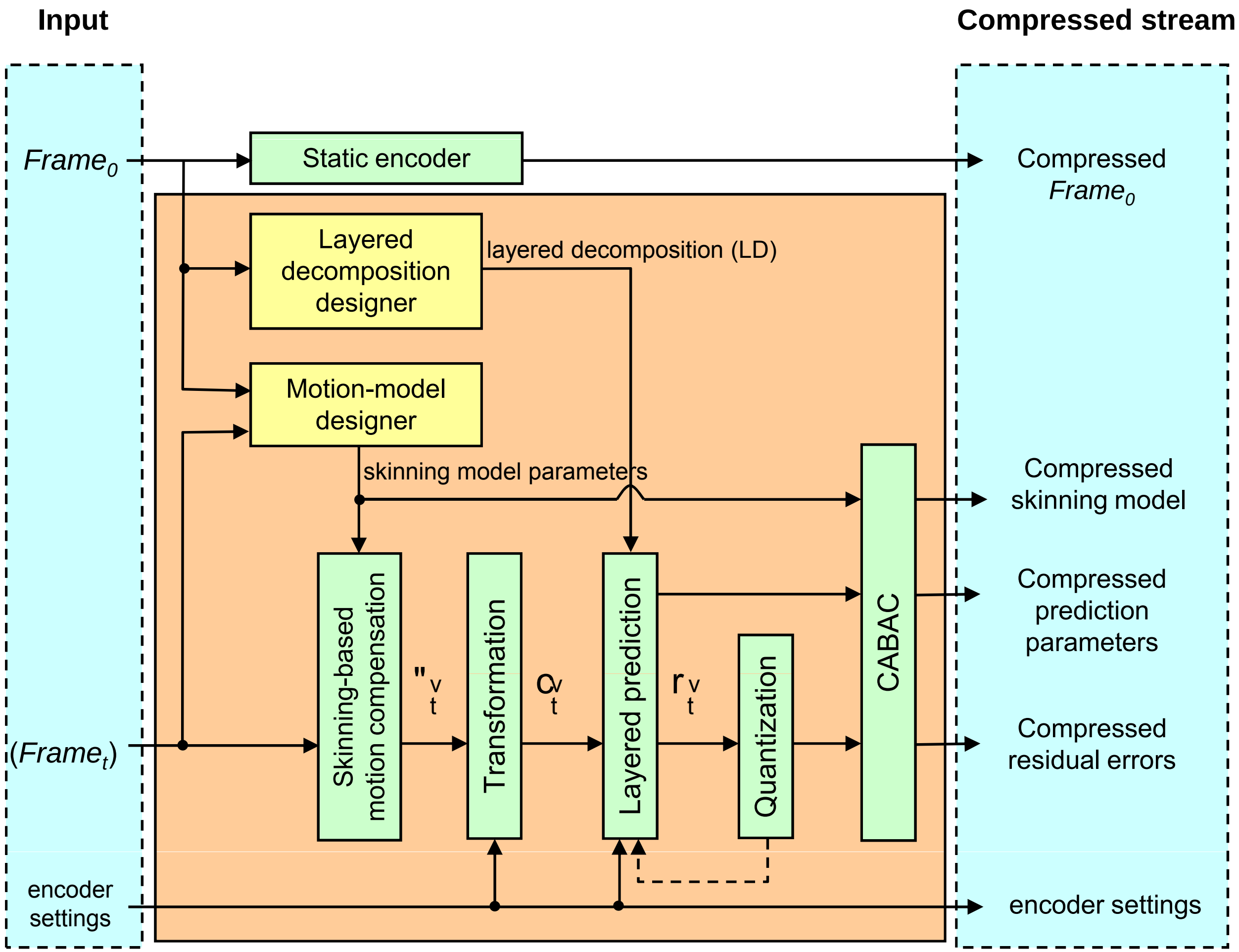
MPEG-4 / FAMC

- **FAMC** (MPEG-4, Part 16, Amd. 2, '07/'08)
- **F**rame-based **A**nimated **M**esh **C**ompression
- Based on LD3DMC and Skinning approach
- Employs CABAC for entropy coding
- Supports several types of scalability
  - spatial, temporal, quality and combinations
- Allows efficient compression

MPEG-4 / FAMC

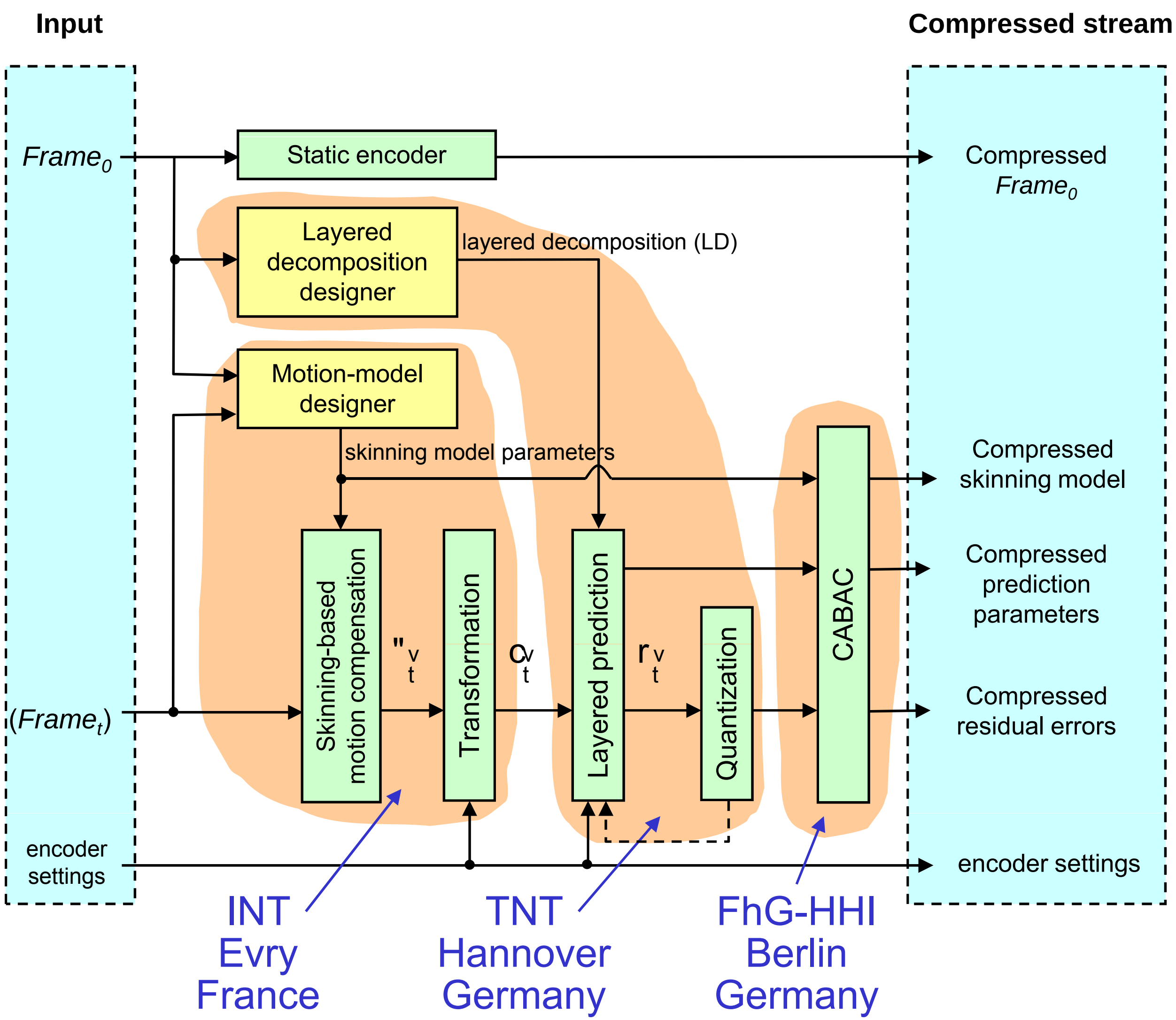

Input
Compressed stream
Frame0
Static encoder
Compressed Frame0
Layered decomposition designer
layered decomposition (LD)
Motion-model designer
skinning model parameters
Compressed skinning model
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed prediction parameters
Compressed residual errors
(Framet)
encoder settings
encoder settings

MPEG-4 / FAMC

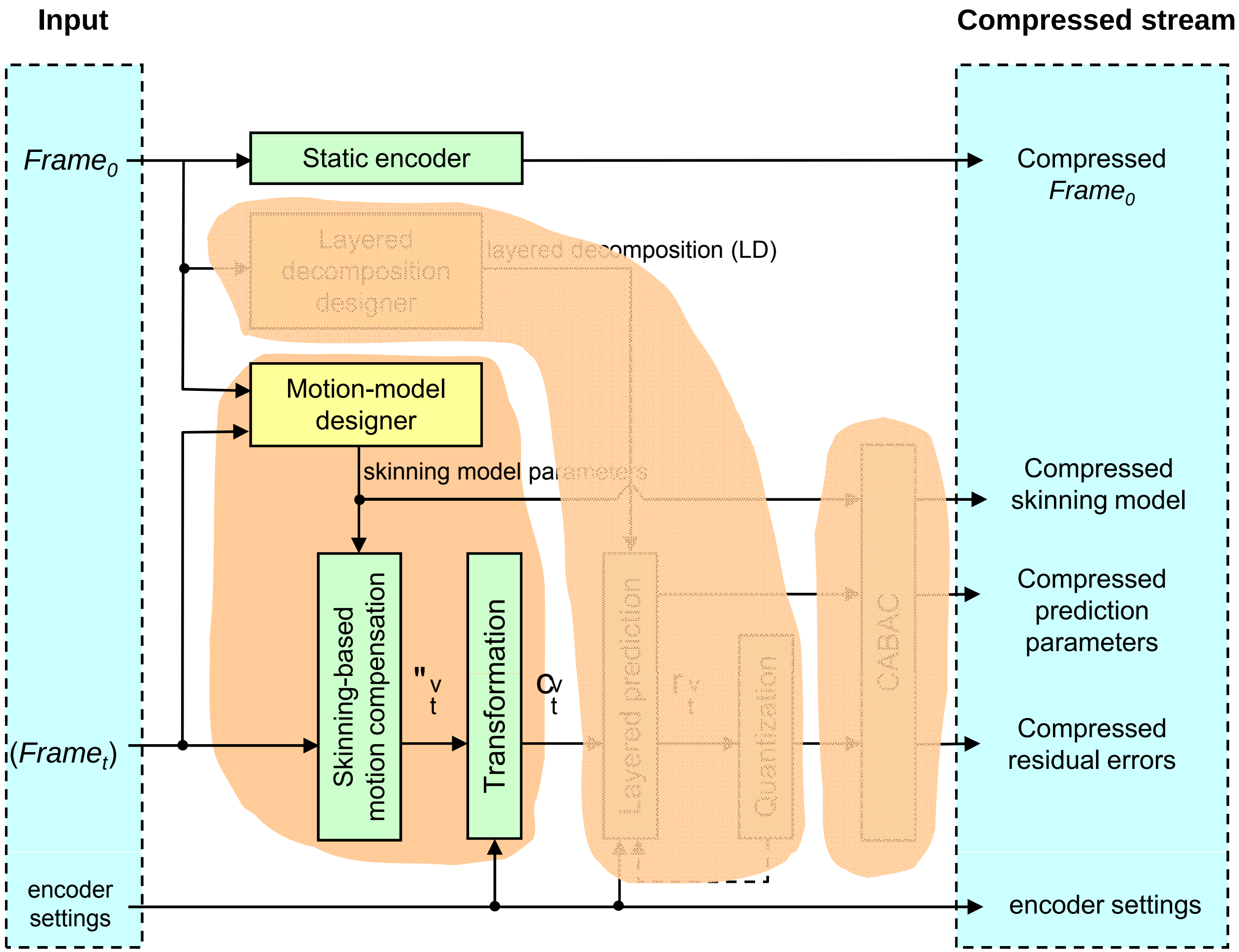


Input
Compressed stream
Frame0
Static encoder
Compressed Frame0
Layered decomposition designer
layered decomposition (LD)
Motion-model designer
skinning model parameters
Compressed skinning model
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed prediction parameters
Compressed residual errors
(Framet)
encoder settings
encoder settings

MPEG-4 / FAMC


Input
Compressed stream
Frame0
Static encoder
Compressed Frame0
Layered decomposition designer
layered decomposition (LD)
Motion-model designer
skinning model parameters
Compressed skinning model
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed prediction parameters
Compressed residual errors
(Framet)
encoder settings
encoder settings
INT Evry France
TNT Hannover Germany
FhG-HHI Berlin Germany

MPEG-4 / FAMC


Input
Compressed stream
Frame0
Static encoder
Compressed Frame0
Layered decomposition designer
layered decomposition (LD)
Motion-model designer
skinning model parameters
Compressed skinning model
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed prediction parameters
(Framet)
Compressed residual errors
encoder settings
encoder settings

MPEG-4 / FAMC

Skinning-based motion compensation:

- Determine K clusters of vertices such that the motion of each cluster *k* at time instance *t* can be accurately described by an affine transform relative to the first frame:

$$\hat{p}_t^v = A_t^k \, p_0^v$$

- Improve motion compensation by using skinning weights w_v^k:

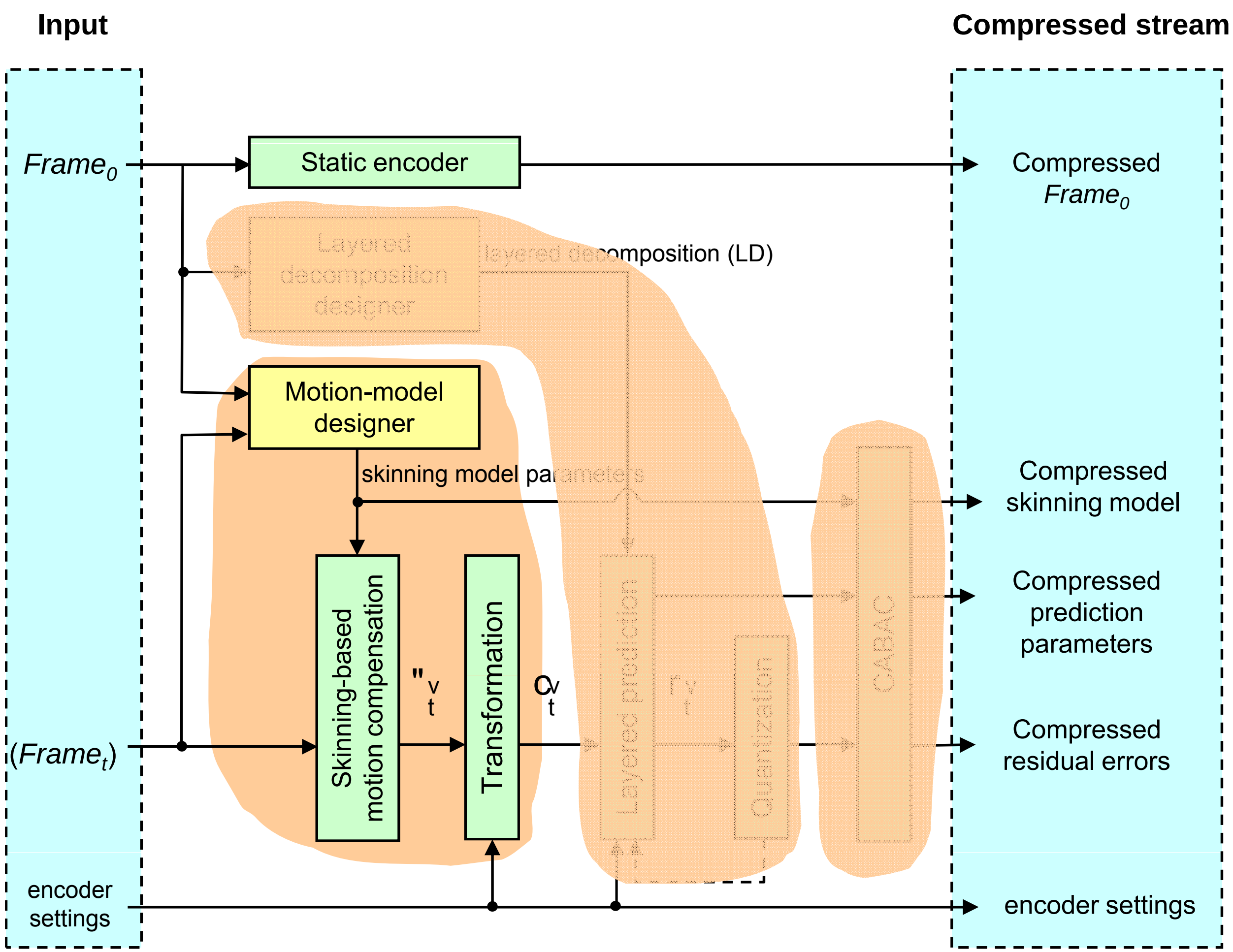
$$\hat{p}_t^v = \left(\sum_{k=1}^{K} w_v^k A_t^k \right) p_0^v$$

- Perform skinning based motion-compensation:

$$\varepsilon_t^v = p_t^v - \hat{p}_t^v$$

- Encode skinning model:
  - all affine transforms A_t^k and
  - all weights w_v^k
- Conduct ε_t^v values to next component
- The determination of A_t^k and w_v^k is subject of analysis of the entire dynamic mesh

MPEG-4 / FAMC

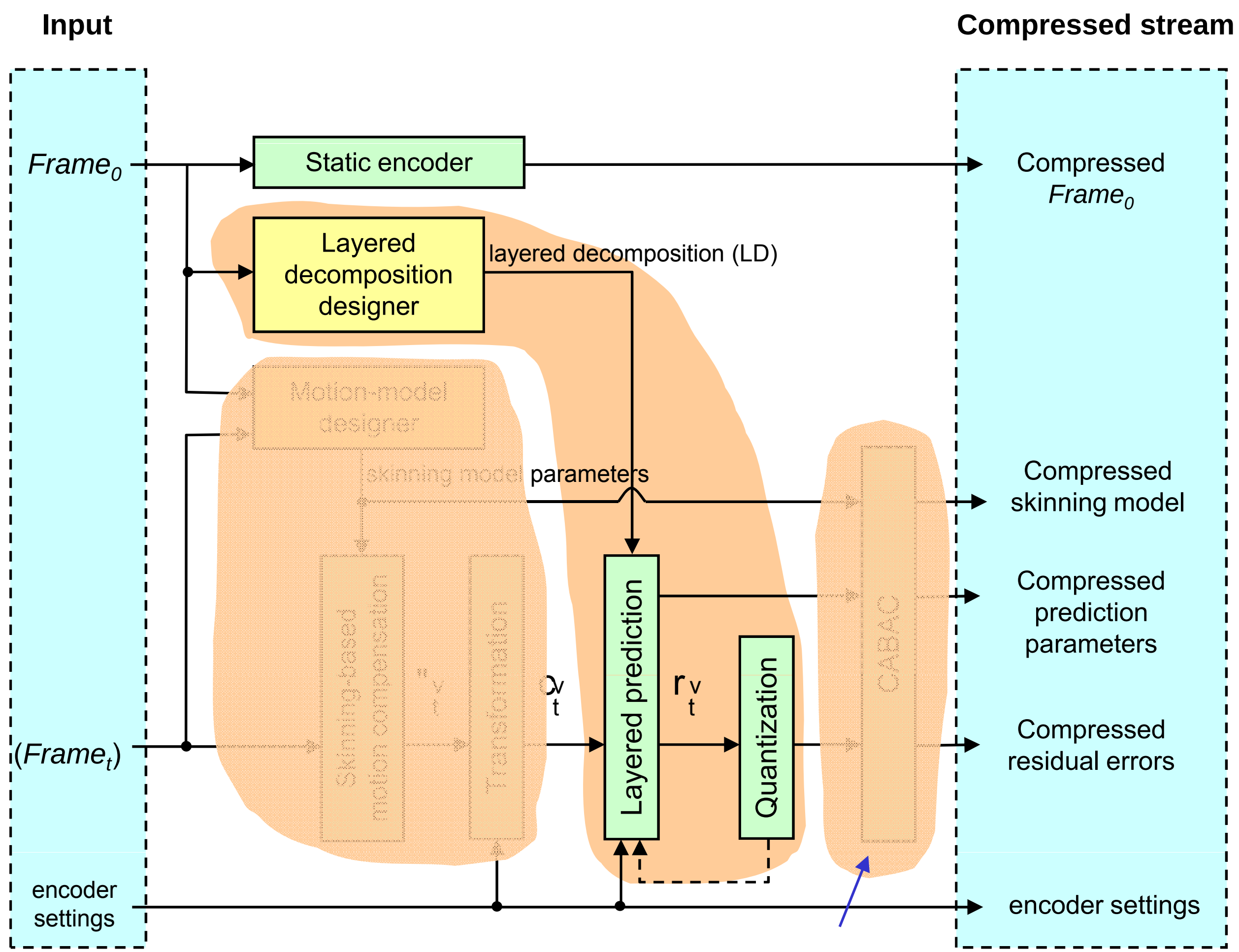


Input
Compressed stream
Frame0
Static encoder
Compressed Frame0
Layered decomposition designer
layered decomposition (LD)
Motion-model designer
skinning model parameters
Compressed skinning model
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed prediction parameters
(Framet)
Compressed residual errors
encoder settings
encoder settings

MPEG-4 / FAMC

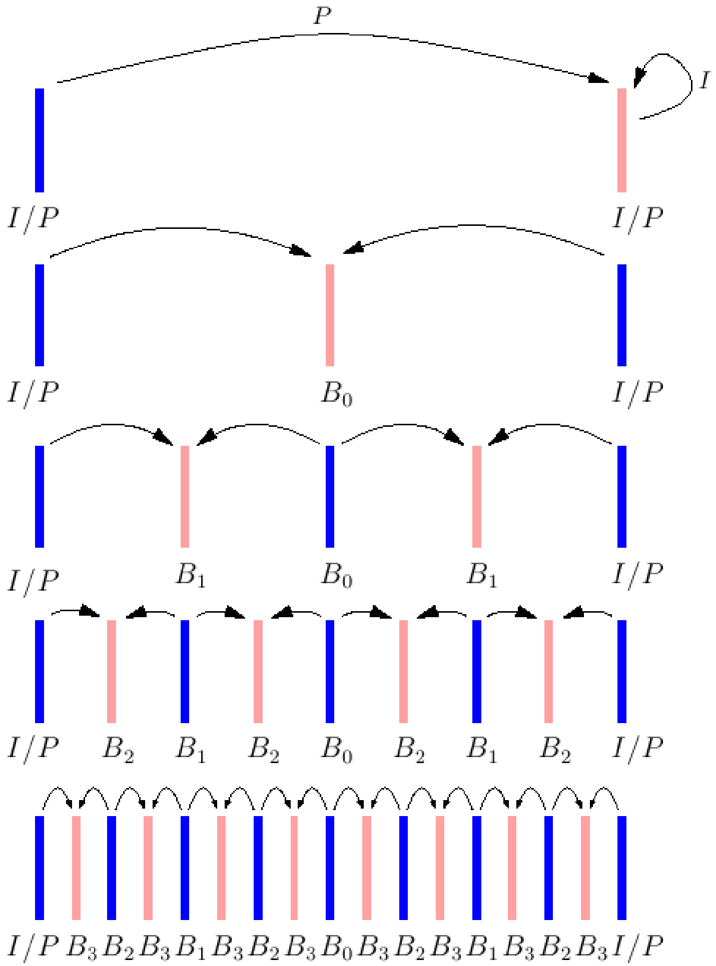
- Transformation of residuals in temporal direction:
  - DCT
  - Lifting Scheme (Wavelet)
  - Bypass

MPEG-4 / FAMC



Input
Compressed stream
Frame0
Static encoder
Compressed Frame0
Layered decomposition designer
layered decomposition (LD)
Motion-model designer
skinning model parameters
Compressed skinning model
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed prediction parameters
Compressed residual errors
(Framet)
encoder settings
encoder settings

MPEG-4 / FAMC



Temporal Scalability

- Hierarchical B-frames
- Frame encoding order is not equal to frame display order
- Partial decoding leads to reduced frame rate

MPEG-4 / FAMC

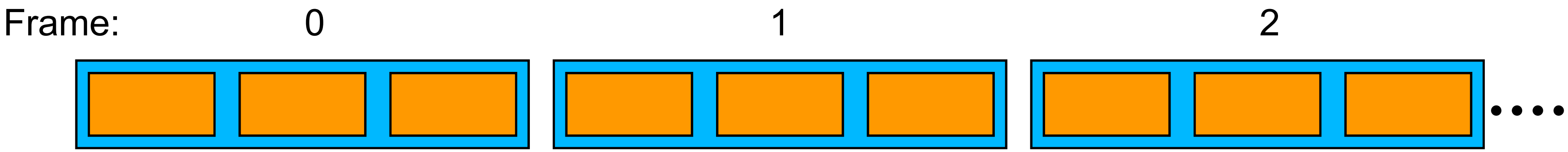
Prediction parameters encoded per frame:

- frame number
- frame type
  - I-frame (no reference frames)
  - P-frame (one reference frame)
  - B-frame (two reference frames)
- frame numbers of reference frames
- applied prediction type (delta, linear, non-linear)

MPEG-4 / FAMC

Spatial scalability

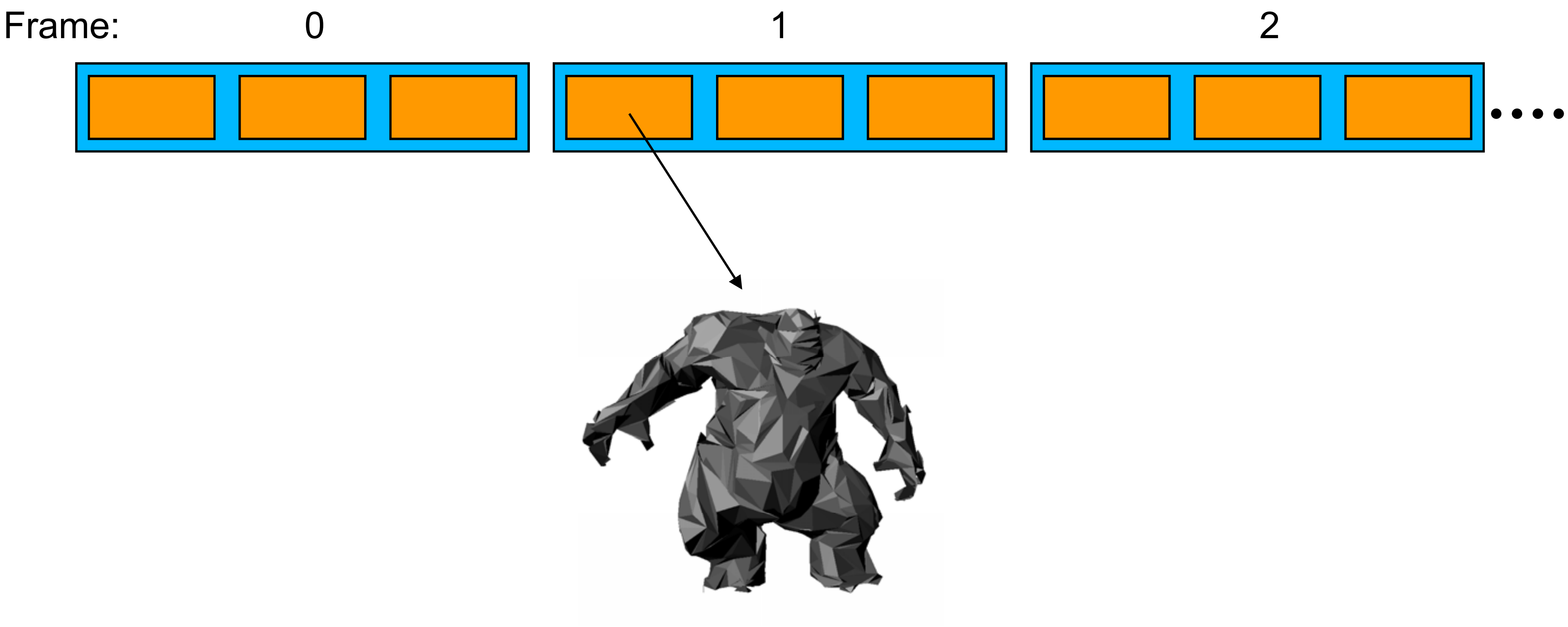
- Spatial layers allow frame-wise decoding with successively increasing spatial resolution



MPEG-4 / FAMC

Spatial scalability

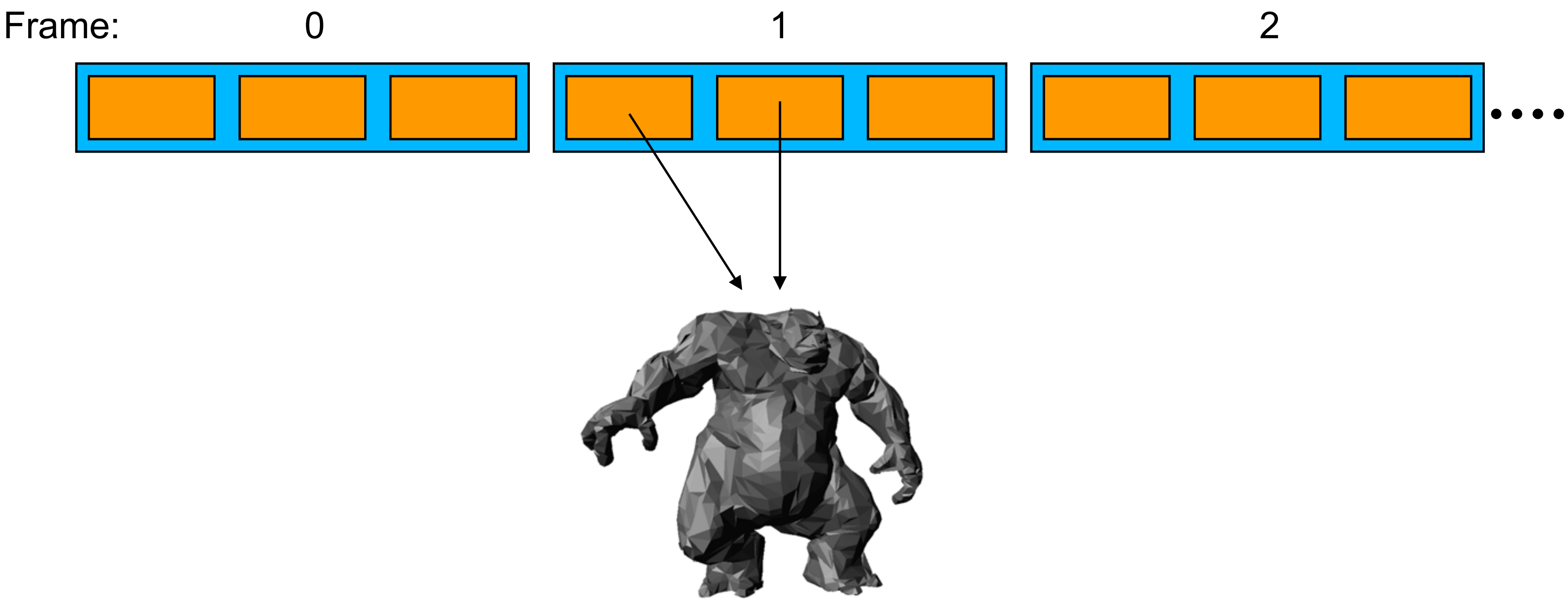
- Spatial layers allow frame-wise decoding with successively increasing spatial resolution



MPEG-4 / FAMC

Spatial scalability

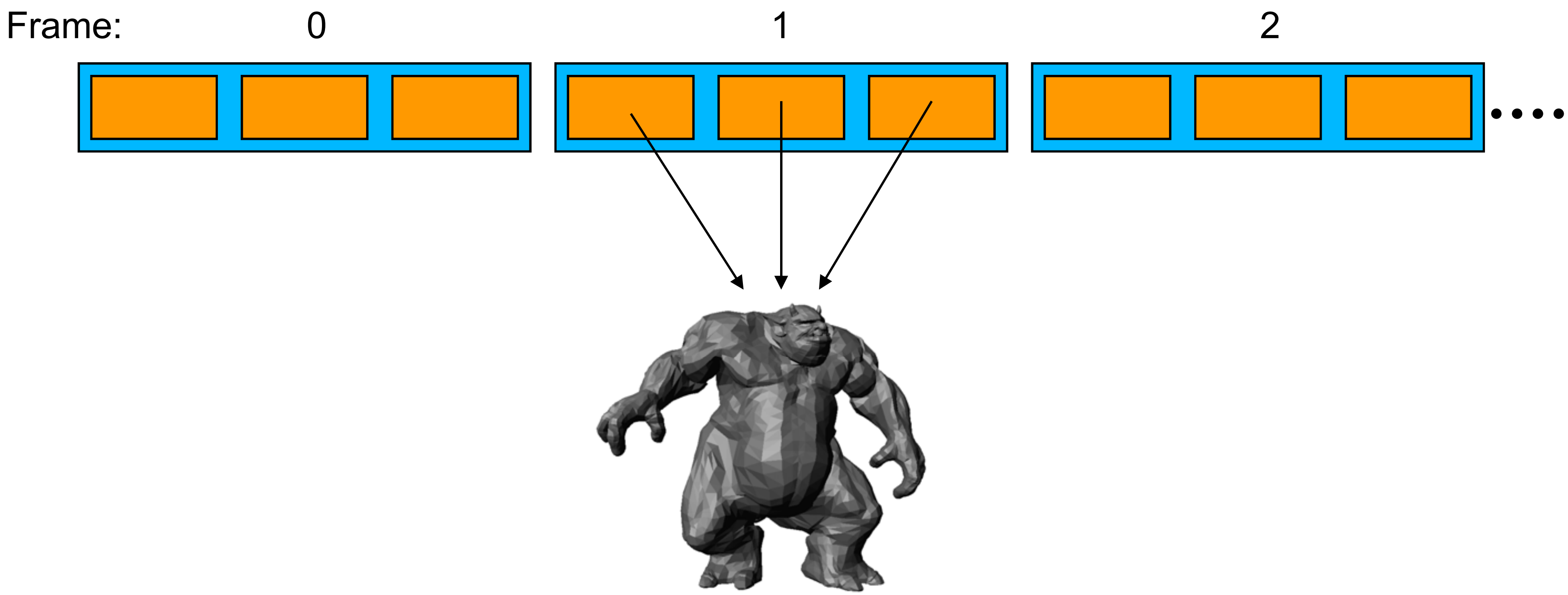
- Spatial layers allow frame-wise decoding with successively increasing spatial resolution



MPEG-4 / FAMC

Spatial scalability

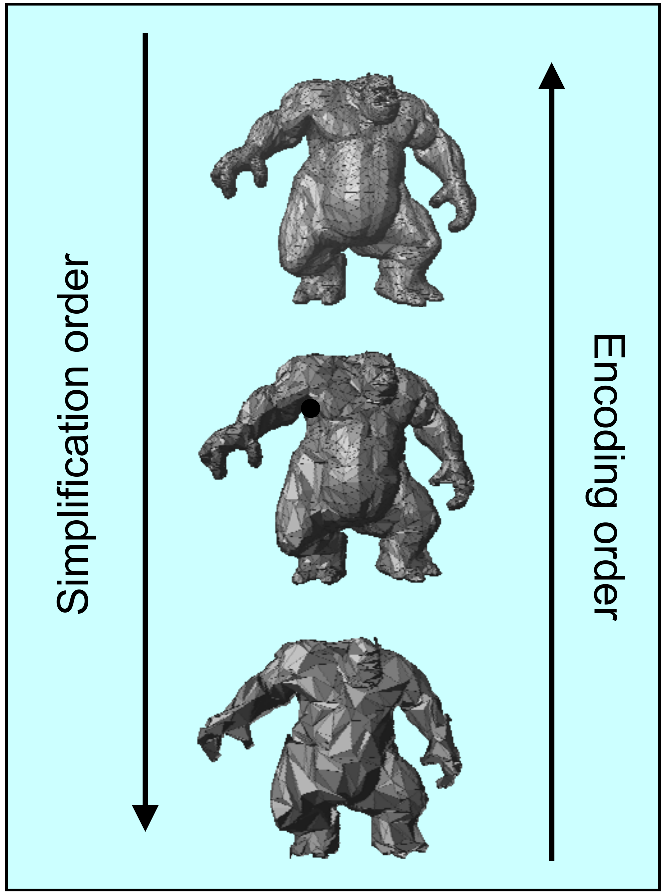
- Spatial layers allow frame-wise decoding with successively increasing spatial resolution

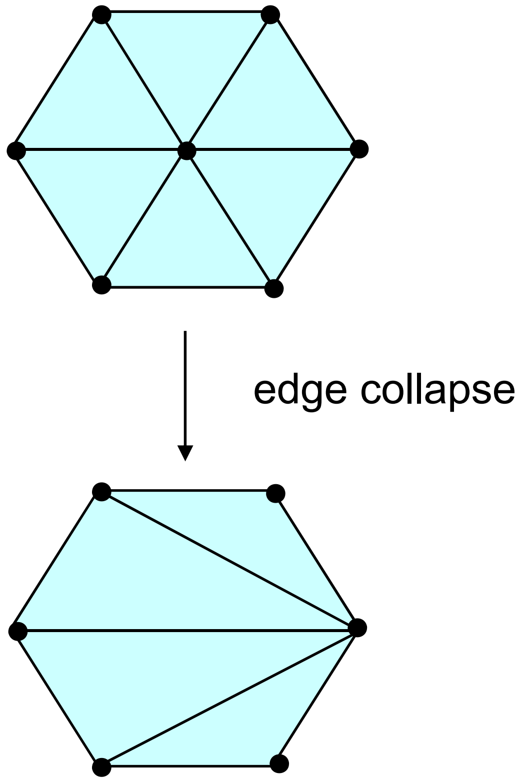


MPEG-4 / FAMC

Layered Prediction

- Determine sequences of mesh simplification operations (edge collapse) based on first frame => progressive mesh representation

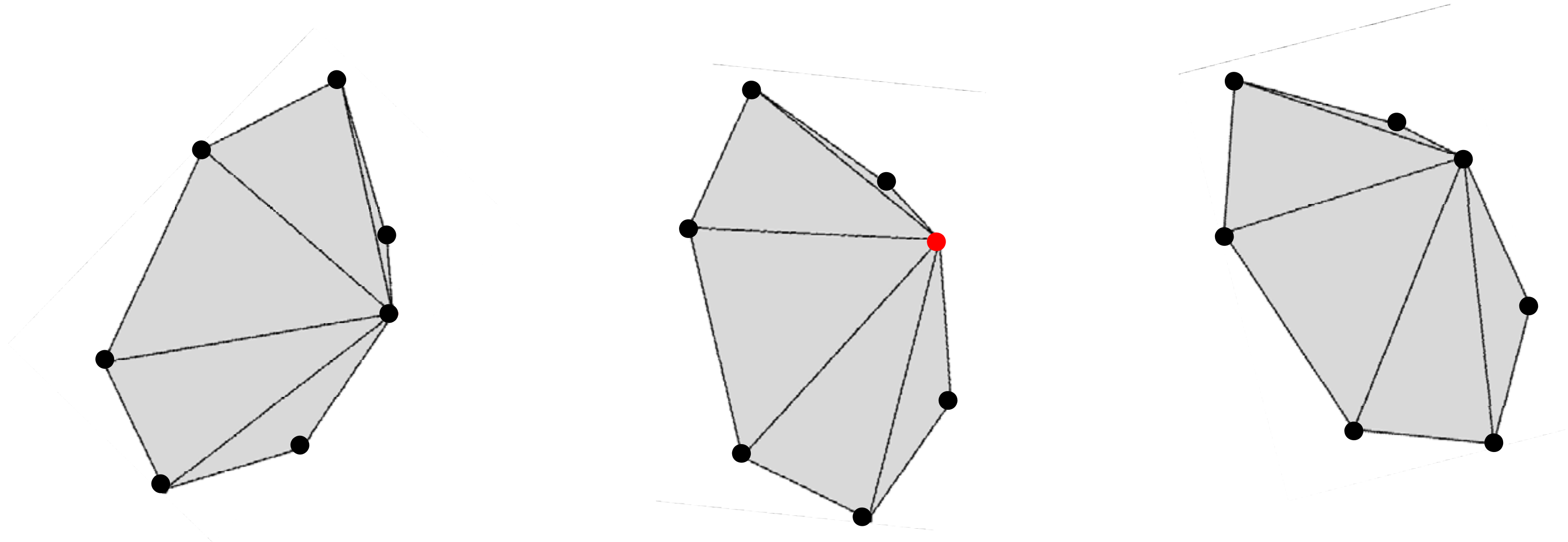




MPEG-4 / FAMC

- **Linear predictor**

reference frame 1 | current frame | reference frame 2

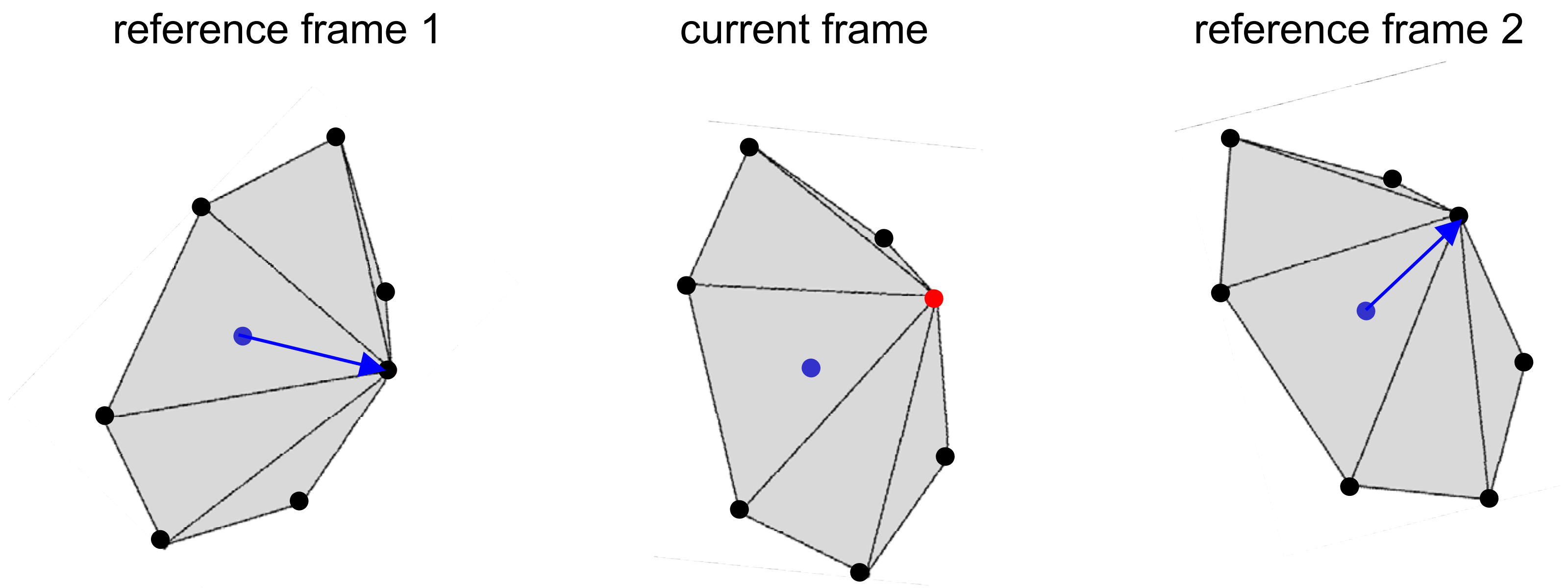


● encoded vertex locations

● vertex location to be encoded

MPEG-4 / FAMC

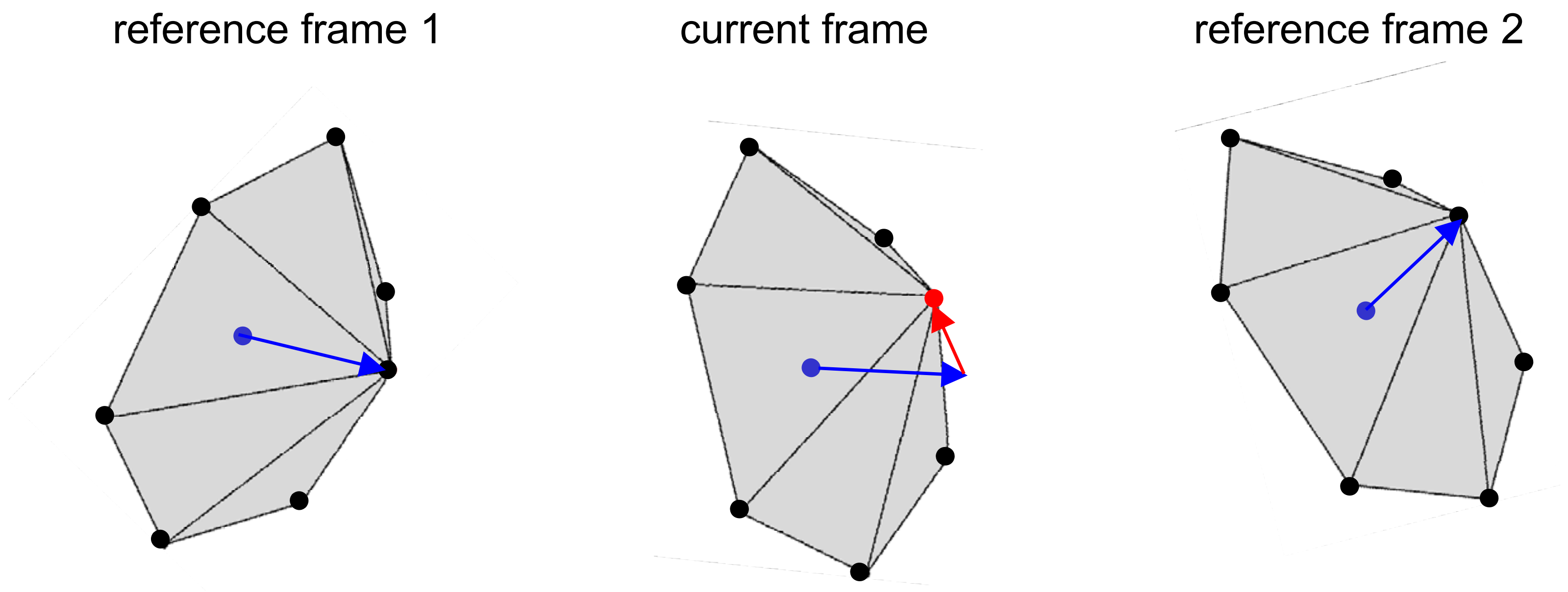
- **Linear predictor**



- encoded vertex locations
- vertex location to be encoded
- barycenter of one-ring

MPEG-4 / FAMC

• **Linear predictor**

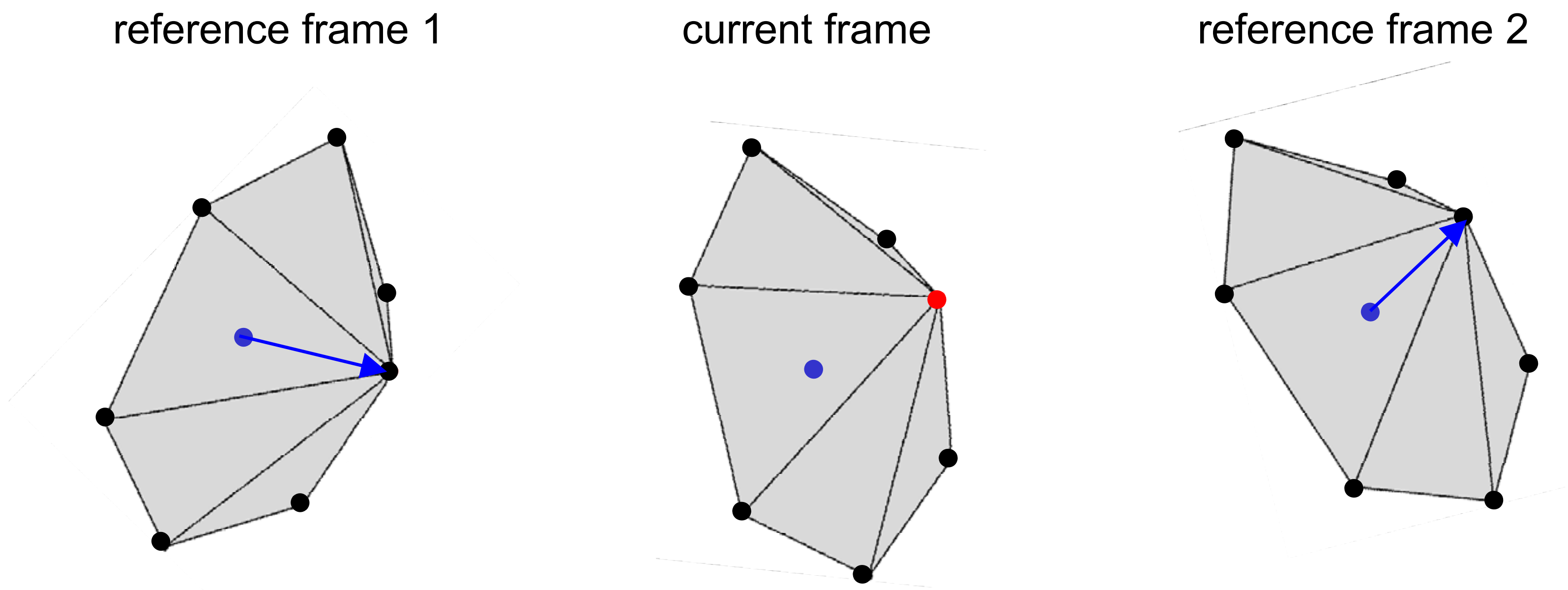


● encoded vertex locations
● vertex location to be encoded
● barycenter of one-ring

↑ is entropy encoded

MPEG-4 / FAMC

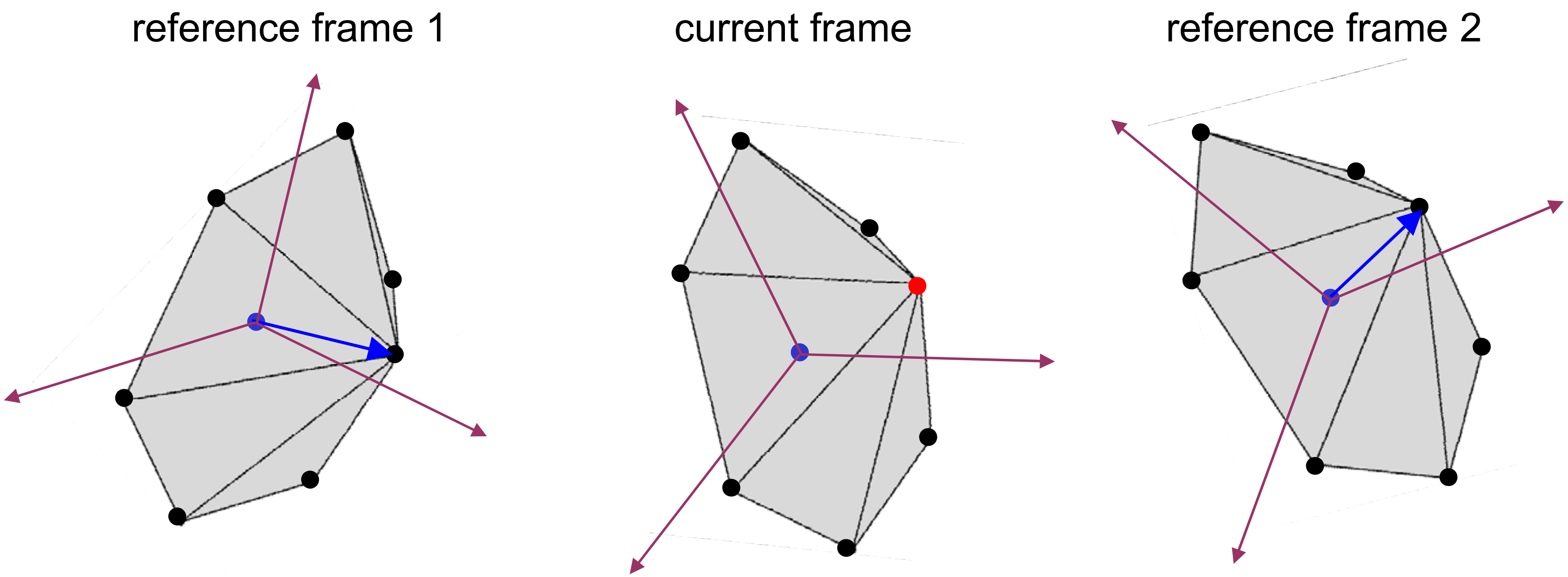
- **Non-linear predictor**



- encoded vertex locations
- vertex location to be encoded
- barycenter of one-ring

MPEG-4 / FAMC

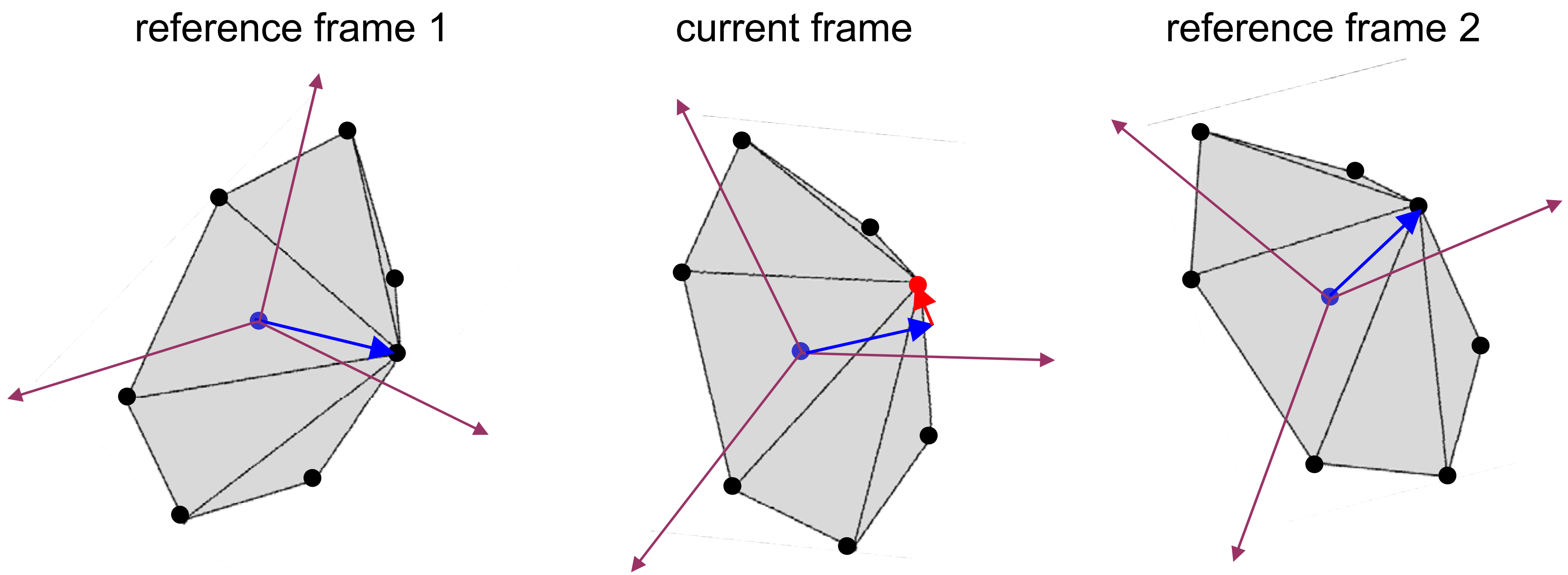
- **Non-linear predictor**



- encoded vertex locations
- vertex location to be encoded
- barycenter of one-ring

MPEG-4 / FAMC

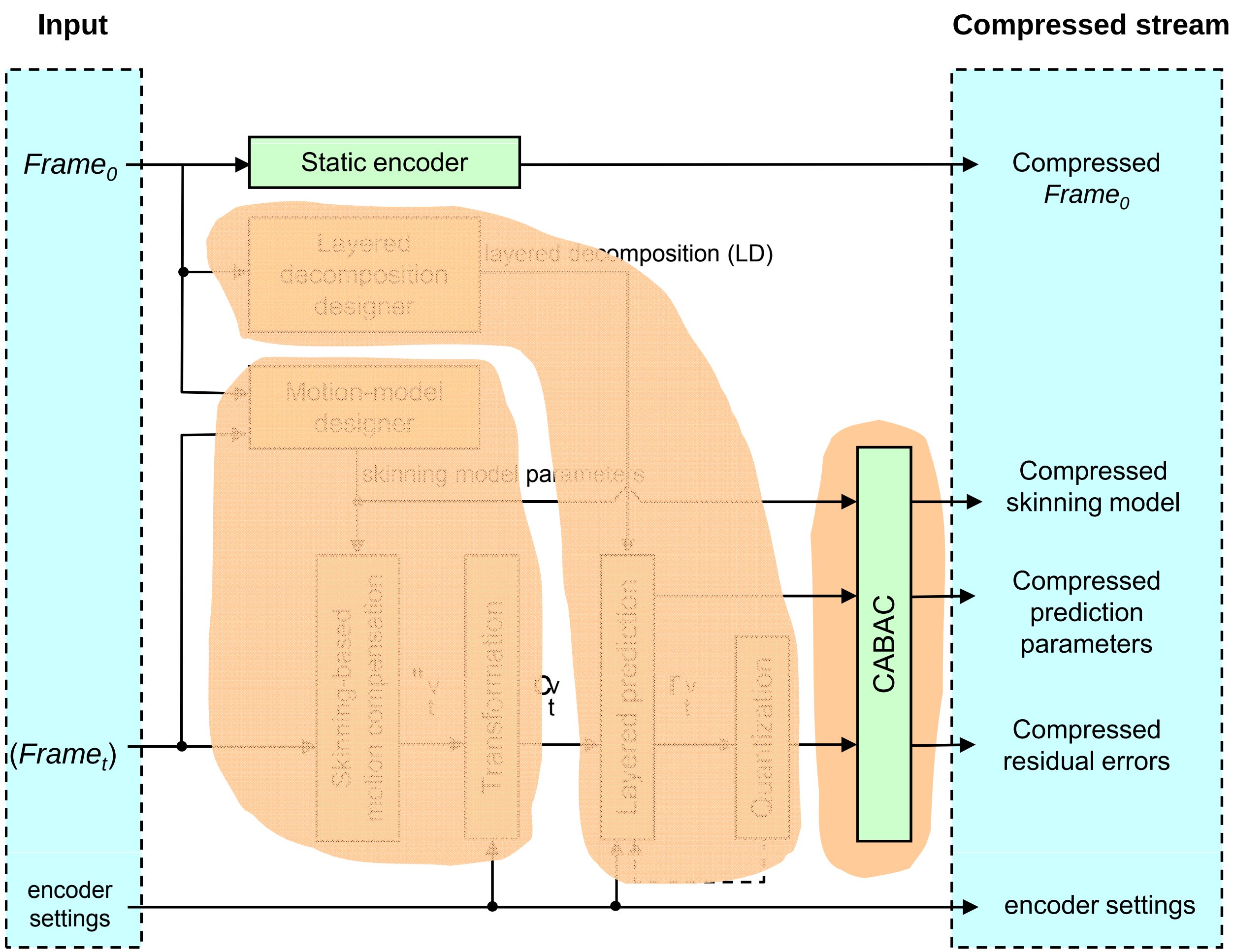
- **Non-linear predictor**



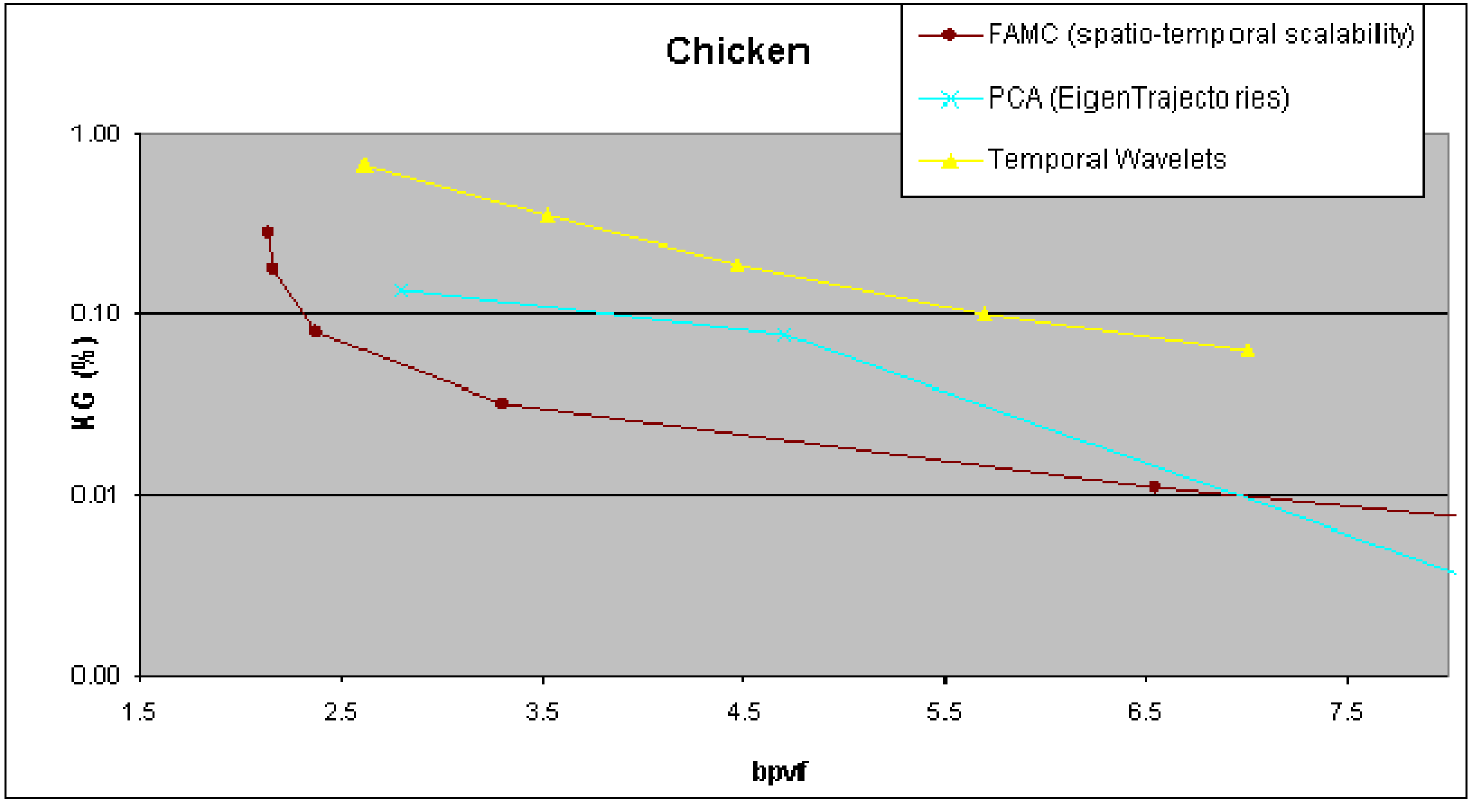
- ● encoded vertex locations
- ● vertex location to be encoded
- ● barycenter of one-ring

– error free prediction in the case of rigid motion

MPEG-4 / FAMC

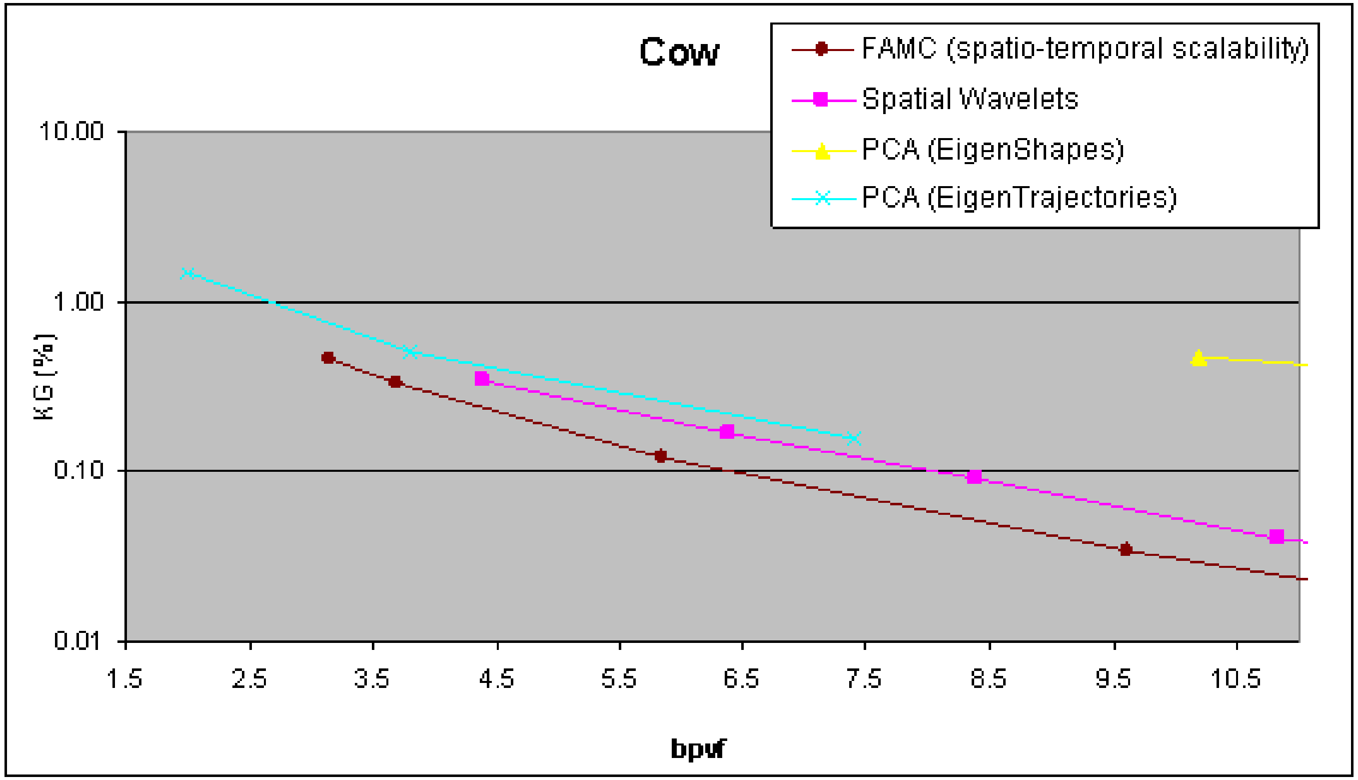

Input
Compressed stream
Frame$_0$
Static encoder
Compressed Frame$_0$
layered decomposition (LD)
Layered decomposition designer
Motion-model designer
skinning model parameters
Skinning-based motion compensation
Transformation
Layered prediction
Quantization
CABAC
Compressed skinning model
Compressed prediction parameters
Compressed residual errors
(Frame$_t$)
encoder settings
encoder settings

MPEG-4 / FAMC



- Chicken sequence
  - fast and slow motion of an articulated object (wings and legs)

MPEG-4 / FAMC



- Cow sequence

  -fast global and local motion (physical deformation)

Support of Scalability

Approaches	Animation level		Frame level	
	spatial	quality	spatial	quality
Predictive vertex traversal based	no	no	no	no
DCT / Wavelets (temporal)	no	yes	no	no
PCA EigenTrajectories	no	yes	no	no
PCA EigenShapes	no	no	no	yes
Predictive simplification based	no	no	yes	no
FAMC	yes	yes	yes	no

Summary and Conclusions

- Very active research area
- Unfortunately, no consensus about good objective error measure
  - KG error, 4D Hausdorf, DA ribbon error (MPEG-4), other?
- Many different approaches
  - PCA and Wavelet based approaches
  - Skinning based approaches
  - Image based approaches
  - Predictive simplification based approaches
- New Amd. to the MPEG-4 standard: FAMC
  - combination of skinning and predictive simplification based approach
  - avg. gain in bit rate of 60% compared to previous MPEG-4 technology
  - gains in bit rate of 20% - 40% compared to PCA and wavelet based approaches

What next?

- Compresson of static mesh geometry
  - gains of 50% using spectral compression (high codec complexity)
  - interest in efficient low complexity coding of static meshes

- Most current approaches encode only connectivity and vertex positions
  - compression of vertex normals, colors, texture coordinates or even multi-view texture coordinates of increasing interest
  - compression of normals, colors and texture coordinates supported by MPEG-4/FAMC

- Application in depth maps coding
  - depth maps used in multi-view video for representation of geometry data
  - coding of single view and multi view depth maps using techniques from compression of dynamic meshes

BACKUP