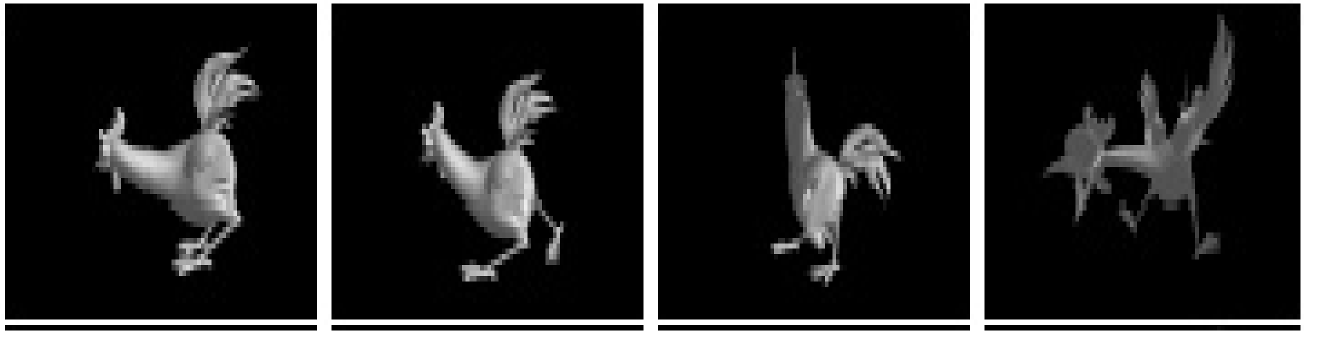
Dynapack: Space-Time compression of the 3D animations of triangle meshes with fixed connectivity

Lawrence Ibarria, Jarek Rossignac
College of Computing and GVU Center
Georgia Tech, Atlanta

Animated meshes

- Dynamic Product inspection
- Virtual Reality Engines
- Interactive 3D movie streaming



Dynapack Overview

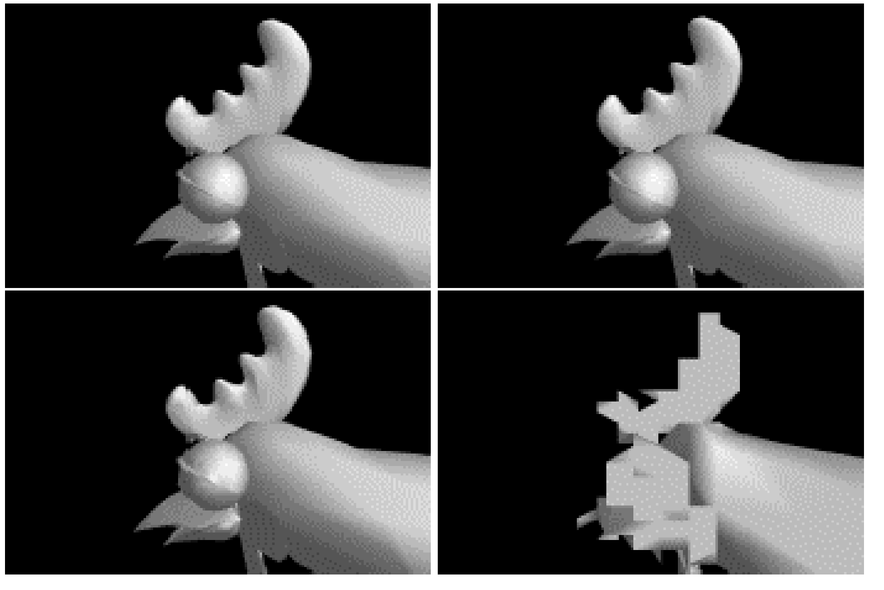
- Quantize dataset to integers
- Encode first Key Frame
- Traverse mesh generating next key-frame
- Proceed with next frame

Quantization

- Quantization is lossy, but not always noticeable

original 32 bits float

quantized to 13 bits
per coordinate



quantized to 11 bits
per coordinate

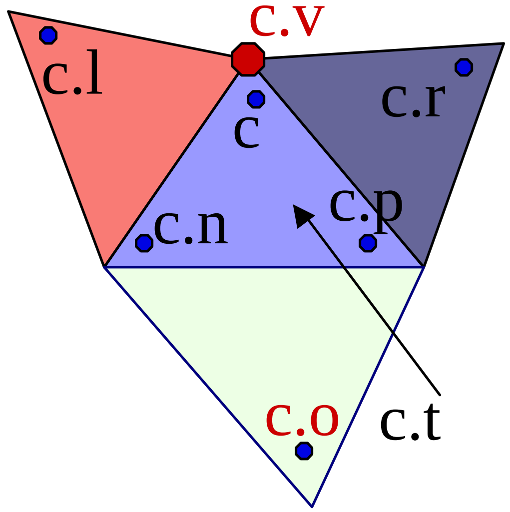
quantized to 7 bits
per coordinate

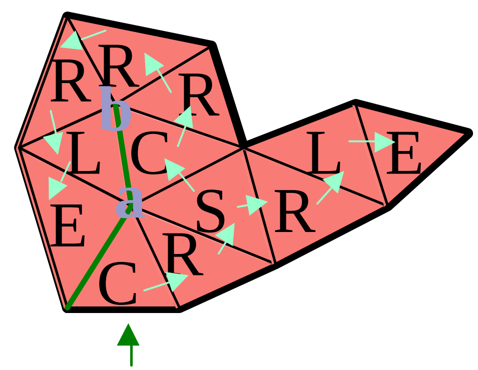
Encoding First Frame

- Connectivity will remain constant through all the animation
- We use Edgebreaker to encode the connectivity
  - Other approaches may easily be used instead
- We use the standard Edgebreaker for the first frame
- We use a modified Edgebreaker for subsequent frames

Dynapack Algorithm (recursive version)

```
dynapack(c) {                                         #compression of a component of a frame
IF c == –1 THEN RETURN;                               #return if a border is reached
IF NOT c.t.m THEN {                                   #if triangle c.t not yet visited
    IF NOT c.v.m THEN {                               #if tip vertex not yet visited
        encode(c.v.g( f ) – predict(c, f ) )          #encode residue coordinates
        c.v.m := TRUE};                               #mark the tip vertex as visited
    c.t.m := TRUE;                                    #mark the triangle as visited
    dynapack(c.r);                                    #try to go to the right neighbor
    dynapack(c.l);}}                                  #try to go to the left neighbor
```



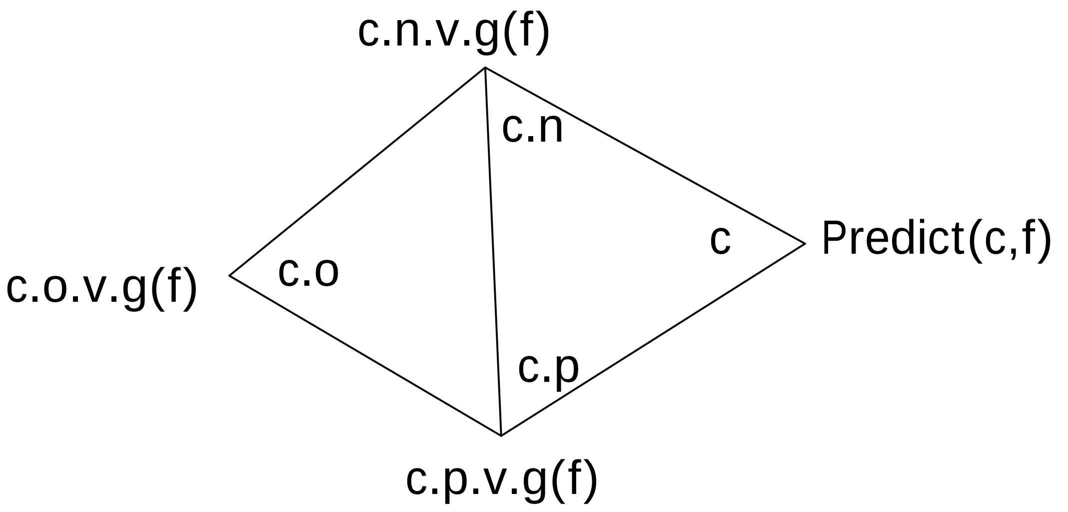


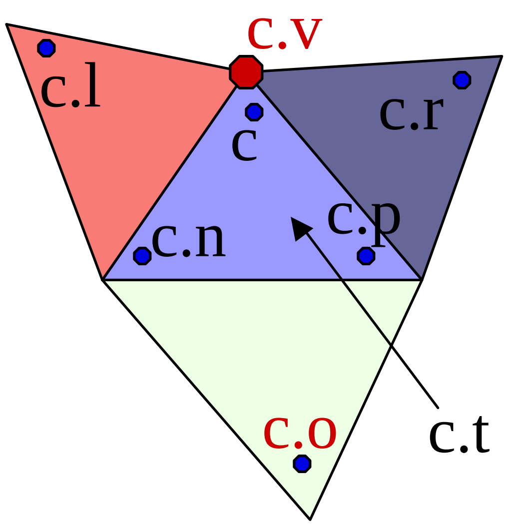
Predictors

- Requirements
  - Simple
  - Small number of points
- Predictors
  - Space only: Parallelogram Predictor
  - Time only: Location on the previous frame
  - Space-Time ELP: Extended Lorenzo Predictor
    - Predicts perfectly translations
  - Space-Time Replica
    - Perfect predictor for translations, rotations, scaling

Space-Only Predictor

- predict(c,f) = c.n.v.g(f)+c.p.v.g(f)–c.o.v.g(f)
  - The old parallelogram
  - Uses one frame at a time
  - Does not exploit temporal coherence





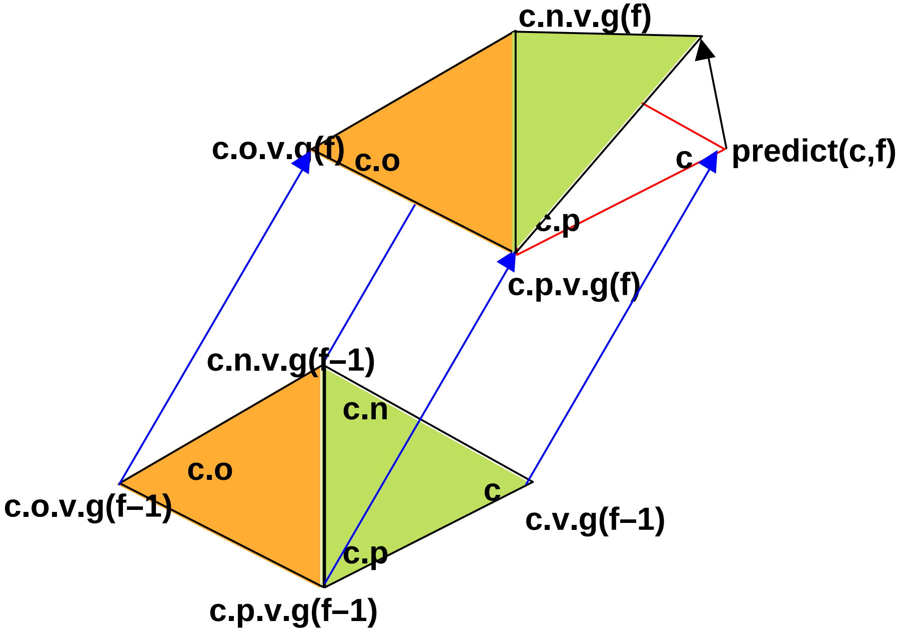
Time-only predictor

- predict(c,f) = c.n.v.g(f-1)
  - Expects the vertex to stay in the same place as in the previous frame
  - Does not exploit coherence between neighbors along the surface

Extended Lorenzo Predictor (ELP)

predict(c,f) = c.v.g(f–1) **+ (** c.n.v.g(f) – c.n.v.g(f–1) **)**
+ (c.p.v.g(f) – c.p.v.g(f–1) **) – (** c.o.v.g(f) + c.o.v.g(f–1) **)**

- Exploits both space and time coherence



Replica Predictor

- ***predict(c,f) = c.o.v.g(f)+aA'+bB'+cC'***
  - Expresses a vertex in coordinate system of neighbor triangle
  - Exact predictor for rigid body transforms and scaling

$$a = \frac{A\cdot D * B\cdot B - B\cdot D * A\cdot B}{A\cdot A * B\cdot B - A\cdot B * A\cdot B}$$

$$b = \frac{A\cdot D * A\cdot B - B\cdot D * A\cdot A}{A\cdot B * A\cdot B - B\cdot B * A\cdot A}$$

$$c = D\cdot\frac{A\times B}{\|A\times B\|^2} * \sqrt{\|A\times B\|}$$

$$A' = c.p.v.g(f) - c.o.v.g(f)$$

$$B' = c.n.v.g(f) - c.o.v.g(f)$$

$$C' = \frac{A'\times B'}{\sqrt[3]{\|A'\times B'\|^2}}$$

$$A = c.p.v.g(f-1) - c.o.v.g(f-1))$$

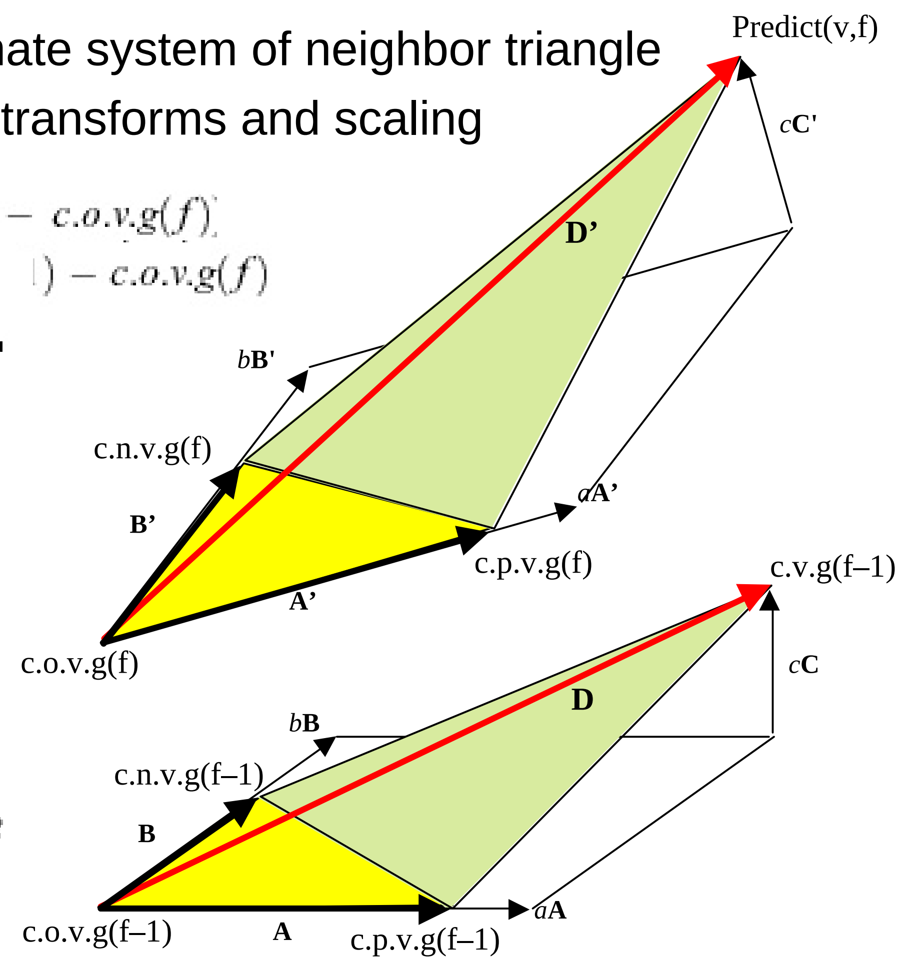
$$B = c.n.v.g(f-1) - c.o.v.g(f-1)$$

$$C = \frac{A\times B}{\sqrt[3]{\|A\times B\|^2}}$$

$$\mathbf{D} = c.v.g(f-1) - c.o.v.g(f-1)$$

$$\text{c.v.g(f-1)} = c.o.v.g(f-1) + aA + bB + cC$$

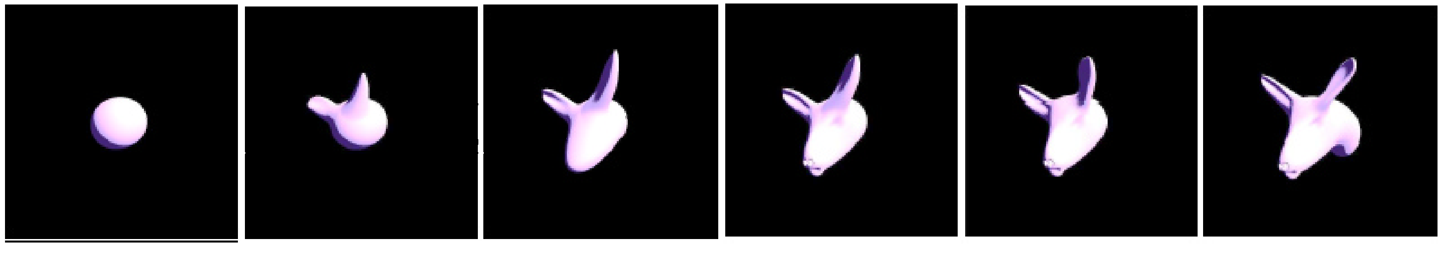
$$D = aA + bB + cC. \longrightarrow \begin{aligned} A\cdot D &= aA\cdot A + bA\cdot B \\ B\cdot D &= aB\cdot A + bB\cdot B. \end{aligned}$$



Results: sub-sampled head

- Space only is very poor
- Other 3 are similar at 13 bits
- Replica is the best for cruder quantization

Head Shaping	7 Bit	9 Bit	11 Bit	13 Bit
Space only	3.07	4.94	6.98	9.16
Time Only	0.80	1.13	1.52	2.02
ELP	0.61	0.96	1.42	2.05
Replica	0.60	0.94	1.39	2.02



Results: Chicken Crossing

- ELP and Replica are much better than the other two
- They yield similar results

Chicken Crossing	7 Bit	9 Bit	11 Bit	13 Bit
Space only	1.90	3.37	5.20	7.19
Time Only	1.78	3.29	5.03	6.91
ELP	1.37	1.79	2.28	3.01
Replica	1.37	1.83	2.35	2.91

Chicken Crossing:

•400 Frames

•31 connected components

•3030 vertices

•5664 triangles

Conclusions

- Effective compression of 3D animation frames
- Trivial implementation
- Needs only previous frame
  - Small foot-print
  - Suitable for out-of-core compression/decompression of large sets
  - Perfect for streaming animations from the Internet
  - Perfect for live compression
- Currently limited to constant connectivity
- Not limited to any kind of deformations