

Harmonic skeleton for realistic character animation

Grégoire AUJAY
Francis LAZARUS

Franck HÉTROUY
Christine DEPRAZ

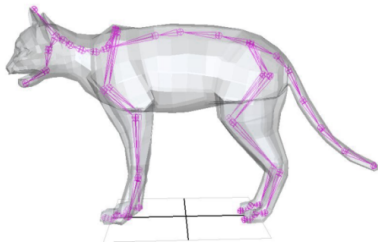


`Franck.Hetroy@imag.fr`

Skeletal animation

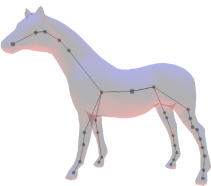
Building an animated 3D character requires to:

- 1 Create a 3D mesh
- 2 Set the 3D position and hierarchy of skeleton joints
- 3 Set the 3D orientation of skeleton joints
- 4 Create high-level controllers for animation (IK, constrains)
- 5 Define skinning weights



Skeleton computation

- Nowadays:
 - By hand
 - By experts
- Automatic methods:
 - Little control over the result
 - Provide noisy skeletons (unwanted joints)
 - Rely only on the geometry of the shape
 - ↪ *not the anatomy of the model (bone structure)*



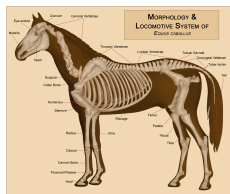
[Liu PG'03]



[Lien SPM'06]

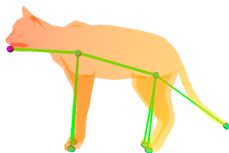


[Tierny PG'06]



Our idea

Control of the **extrema** thanks to a harmonic function

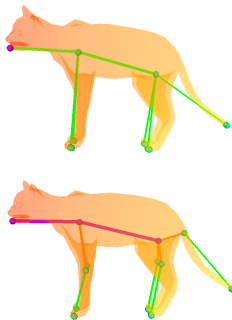


Our idea

Control of the **extrema** thanks to a harmonic function



Automatic association of **semantic information** to nodes/joints



Our idea

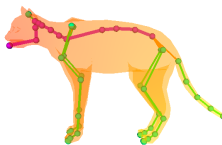
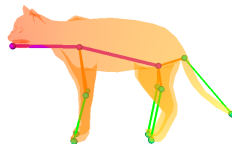
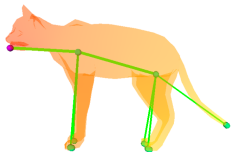
Control of the **extrema** thanks to a harmonic function



Automatic association of **semantic information** to nodes/joints



Adapted refinement to match handmade IK skeletons



Our idea

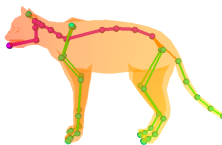
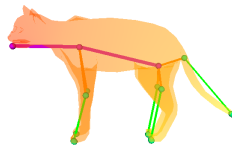
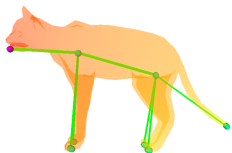
Control of the **extrema** thanks to a harmonic function



Automatic association of **semantic information** to nodes/joints



Adapted **refinement** to match handmade IK skeletons

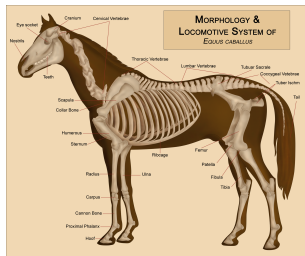
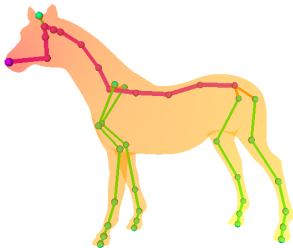


~> *Help for both experts and non-experts*

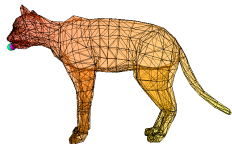
Contributions

A skeleton generation algorithm with the following properties:

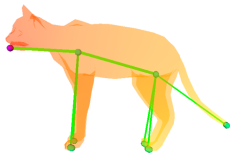
- **Semantic decomposition** of both skeleton and shape
- **Specialized heuristics** for bipeds and quadruped mammals
- **Robustness** (pose+deformation: see paper)
- **Fast and almost no user intervention**, but control is possible



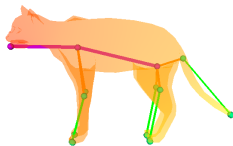
Algorithm overview



Mesh
+ source vertex



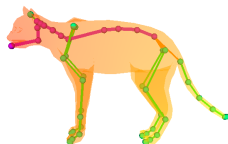
Harmonic
graph



Harmonic
skeleton



A priori
knowledge

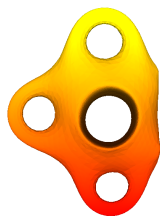


Adapted
refinement

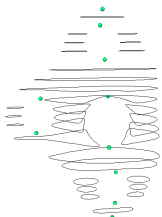
Reeb graph

Reeb graph of $f = M / \sim$

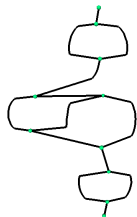
$$x_1 \sim x_2 \iff \begin{cases} f(x_1) = f(x_2) \\ \text{and } x_1 \text{ and } x_2 \text{ belong to the same} \\ \text{connected component of } f^{-1}(f(x_1)) \end{cases}$$



mesh



level sets



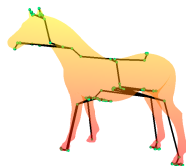
Reeb graph

Harmonic function

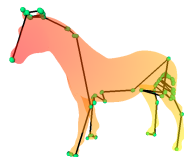
Laplace equation with boundary conditions:

$$\begin{cases} \Delta f(x) = 0 & \forall x \in M \\ f(x) = g(x) & \forall x \in B_M \end{cases}$$

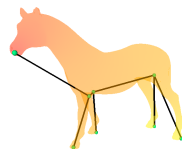
- B_M = set of vertices: **extrema** of the graph
- $g(x)$ = approx. geodesic distance to source vertex



height



distance to source

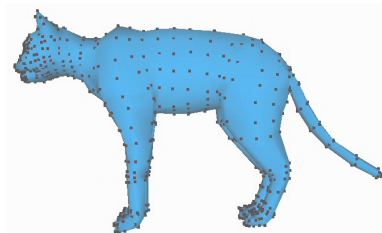


harmonic

Harmonic graph computation

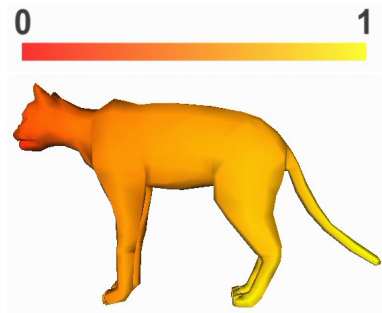
1 Extrema selection

- Automatic or not
- Source vertex on the head (by hand)



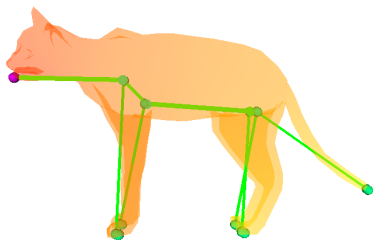
Harmonic graph computation

- 1 Extrema selection
- 2 Harmonic function f computation



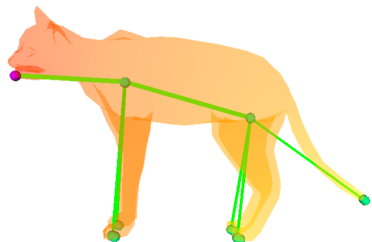
Harmonic graph computation

- 1 Extrema selection
- 2 Harmonic function f computation
- 3 Reeb graph computation
 - In $O(n \log n)$ time
[Cole McLaughlin SoCG'03]

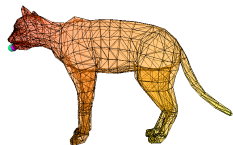


Harmonic graph computation

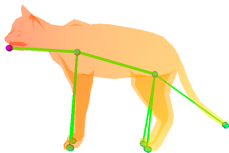
- 1 Extrema selection
 - 2 Harmonic function f computation
 - 3 Reeb graph computation
 - 4 Graph filtering
 - Small (for f) edges removal
 - Related to **persistence**
[Edelsbrunner FOCSS'00]
 - Can be done automatically
[Pascucci SIGGRAPH'07]
- ⇒ *morphology recovered*



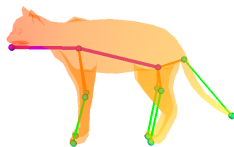
Algorithm overview



Mesh
+ source vertex



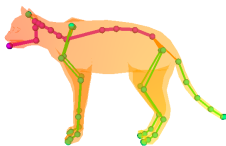
Harmonic
graph



Harmonic
skeleton



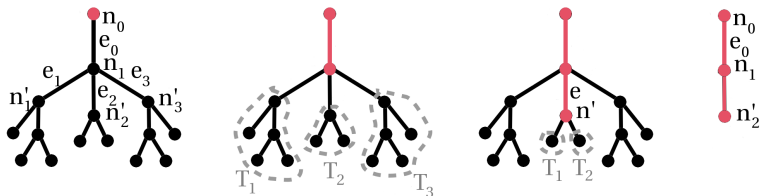
A priori
knowledge



Adapted
refinement

Symmetry axis detection

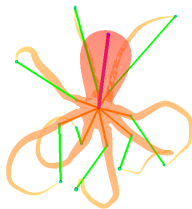
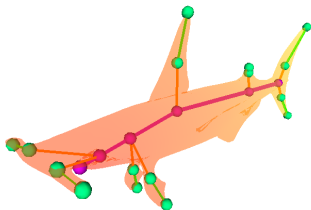
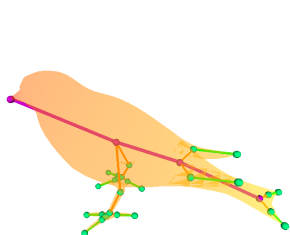
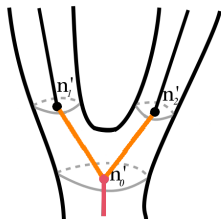
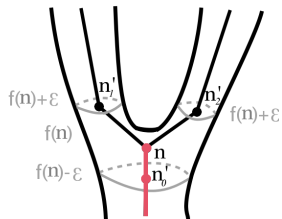
- Symmetry of the **morphology** (2 arms, 2 feet, ...)
- Finding symmetries on a graph: **NP-complete**
- Assumptions:
 - **Source node** on the symmetry axis
 - **No cycle**
 - 2 subtrees **isomorphic** \iff same depth + same degree for root nodes



Simple embedding

Not straightforward
solution for **saddle nodes**

- Insert **regular nodes**
- Use the **symmetry axis**



Joint hierarchy and orientation

Hierarchy:

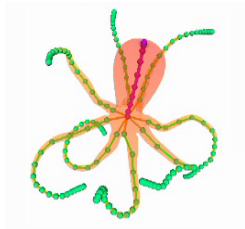
- Use symmetry axis: e.g. last joint = root joint

Orientation:

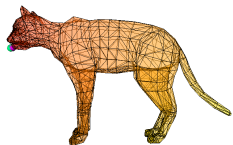
- Use the direction of the embedding of the symmetry axis

Skeleton refinement:

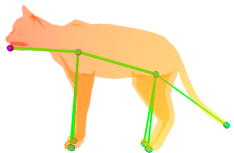
- Add regular nodes on the graph
- Embedding: center of mass of their connected component for f



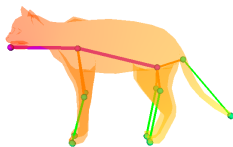
Algorithm overview



Mesh
+ source vertex



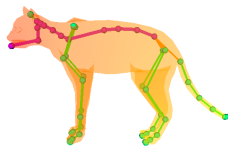
Harmonic
graph



Harmonic
skeleton



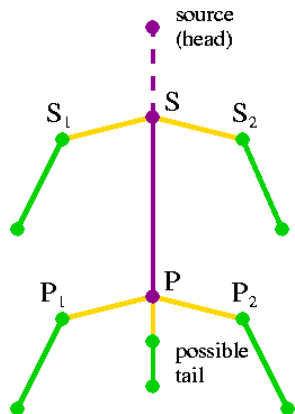
A priori
knowledge



Adapted
refinement

Bipeds and quadrupeds

The symmetry axis gives **semantic information!**



New nodes inserted:

- Edges subdivided
- New extremum for the jaw (biped)/the top of the head (quadruped)

Reference frame

$$\left(Spine, \frac{P_1 P_2}{\|P_1 P_2\|}, Spine \times \frac{P_1 P_2}{\|P_1 P_2\|} \right)$$

Heuristics

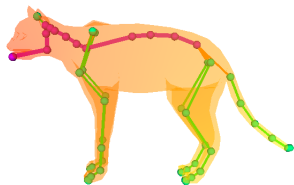
- Some nodes are **shifted** along one of the directions of the reference frame
- New nodes:
 - mean Euclidean position
 - or mean value for f
 - or fitting some **geometric features** [Tierny PG'06]

Heuristics

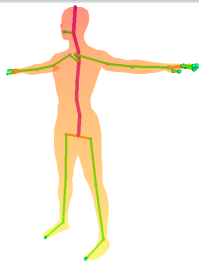
- Some nodes are **shifted** along one of the directions of the reference frame
- New nodes:
 - mean Euclidean position
 - or mean value for f
 - or fitting some **geometric features** [Tierny PG'06]

Example: spine joints, quadruped case

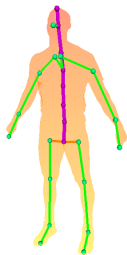
- SP divided into 4 edges
- All joints lifted up along $Spine \times \frac{P_1 P_2}{\|P_1 P_2\|}$
- $J_{new} = J_{simple} + c \cdot \|J_{simple} B\| Spine \times \frac{P_1 P_2}{\|P_1 P_2\|}$
- $c = 0.8/0.76/0.72/0.64$, S so that $Spine$ not modified



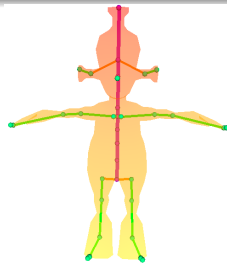
Results



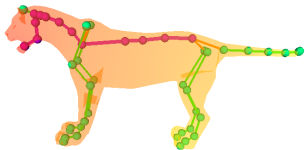
14,000 faces, 0.8s



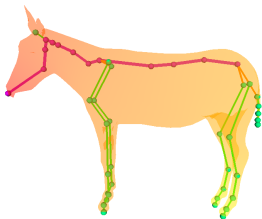
296,000 faces, 36s



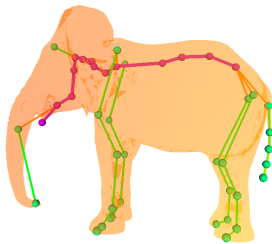
850 faces, 0.05s



4,000 faces, 0.2s



7,000 faces, 0.6s



38,000 faces, 3s



Video: use for animation



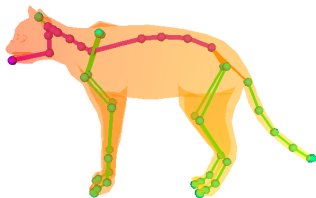
Conclusion

Input:

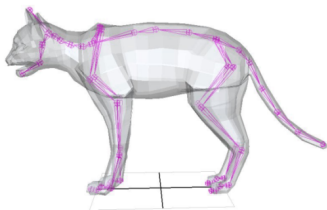
- Manifold **mesh**
- Source vertex (on the head)
- **A priori knowledge** about character's real anatomy

Output:

- IK skeleton closely matching handmade skeleton
- **Semantic information** attached to joints (symmetry axis, anatomy)



Computed



Handmade (previous)

Future work

- **Enhancement**
 - Robust feature extraction [*Zhang ToG'05*]
 - Reeb graph for non-manifold meshes [*Pascucci SIGGRAPH'07*]

- **Skinning**
 - Use f values : vertices \leftrightarrow joints
 - Use joint **semantics** to define adapted weights

Questions?

