Notes

- Questions?
- Assignment 1 should be ready soon (will post to newsgroup as soon as it’s out)

Gauss-Newton

- Idea: nonlinear least-squares is hard, but linear least-squares is easy
- So replace the nonlinear function $x(\theta) - x_{\text{target}}$ with a linear approximation:
  
  $J(\theta - \theta_k) + x(\theta_k) - x_{\text{target}}$

- Then solve the linear least-squares problem to get the “Gauss-Newton” direction:
  
  $d_k = (J^TJ)^{-1}J^T(x_{\text{target}} - x(\theta_k))$

- This avoids the scaling problem of Steepest Descent, and is much more efficient
- Problem: need to solve a linear system

Scaled Steepest Descent (SSD)

- Replace the matrix $J^TJ$ we need to invert with something simpler: $\text{diag}(J^TJ)$
- Diagonal matrices are trivial to invert
  - But guard against zero entries!

Evaluating Jacobians

- Simplest approach in code: numerical approximate with a finite difference
  
  $J_{ij} = \frac{\partial x_i}{\partial \theta_j} \approx \frac{x_i(\theta_1, \ldots, \theta_j + \epsilon, \ldots, \theta_n) - x_i(\theta_1, \ldots, \theta_j, \ldots, \theta_n)}{\epsilon}$

- Can also work out derivative analytically by hand (a little painful)
When to stop

- In our case, absolute minimum of f(.) is zero: stop when it’s smaller than some tolerance
- It might be impossible to get to zero, but at the minimum ∇f=0: stop when |∇f| is small enough
- Or give up when maximum number of iterations reached

Character Rigging

- A “rig” is a model together with a UI for posing it
- At its simplest, a skeleton with joint angles available for motion curves
- May simplify DOF by enforcing relationships between joints
  - E.g. hand and fingers
- May define standard poses (especially for facial expressions!) that can be mixed together
  - Then can set sliders to, say, 70% happy, 20% surprised, ...
  - Take weighted linear combination of pose angles

Breaking Rigs

- Who said animated figures had to have rigid parts?
  - Remember animation principles: stretch & squash, exaggeration, etc.
- Often attractive to break up a rigid skeleton into separate parts (e.g. torso, arms, legs, head)
  - Allow connecting links to change dimension as needed
  - Kinematics only done on a small part - artist doesn’t need to worry about effect on whole (local vs. global control)
  - “If it ain’t broken, then fix it”

What’s left?

- We now have the basics of animation
- Plan for next while:
  - Rendering animations
  - (Semi-)automatic animation
    - Dynamics for rigid bodies
    - Particle systems
    - Skinning, morphing, blending
    - Motion capture
    - Motion control
Rendering for Film

Compositing

- The action of combining multiple “layers” -- parts of each frame -- into the final shot
  - E.g. background + actors + vfx
  - For vfx-intensive shots, there could be dozens of separate layers

- Handling each layer separately
  - makes the problem simpler,
  - allows better division of labour,
  - and gives flexibility in putting the elements together at the end
    (often the majority of CPU time is spent in compositing!)

Atop

- The simplest (useful) and most common form of compositing: put one image “atop” another
  - Image 1 (RGB) on top of image 2 (RGB)

- For each pixel in the final composite, need to know what RGB value to take
  - Where image 1 is opaque, choose RGB₁
  - Where image 1 is “empty”, choose RGB₂
  - Where image 1 is partly transparent, or where image 1 only covers part of the pixel?