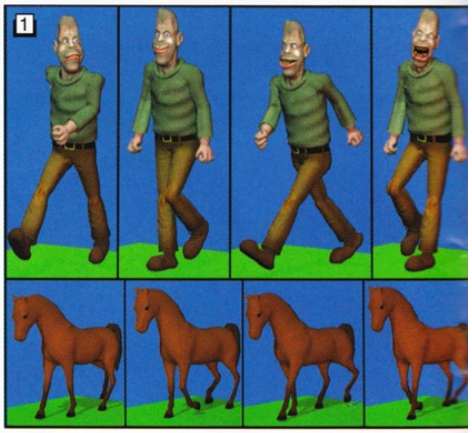
Six-Legged Walk Cycling

Using Imagine for Windows

By R. Cory Collins • Images by the Author

While working on a recent project with Imagine for Windows 1.47, I came upon an interesting challenge. The script in hand called for a pair of six-legged creatures that were to be animated and later composited with the live action. The challenge was to set up multi-legged motion cycles for the creatures.

I have animated both two- and four-legged characters in the past, but this was to be my first effort with six-legged motion cycling. Some of the techniques for bipeds and quadrupeds apply, but there are also some distinct differences. Basically, a biped walks by moving one foot in front of the other, shifting weight to cause the body's center of gravity to move forward. The next foot then swings upward and forward to balance the weight and, finally, propel the body forward again. A quadruped motion cycle is animated in the same manner, only with two sets of legs, with the motion between the front and rear legs offset about half a cycle (Figure 1).



A six-legged creature uses a "double tripod" method of walking. Two sets of three opposing legs form tripods on which the creature maintains balance (Figure 2). This method allows a much sturdier stance than with two or four legs because, at any moment during the walk cycle, a six-legged creature has at least one tripod of legs in contact with the ground.

I decided to build up the motions in layers, first posing the legs, then posing the body itself. I used inverse kinematics for manipulating the legs initially, then went back and tweaked the poses with forward kinematics. I started with all six legs in their neutral, standing position, and applied IK constraints, as shown in Figure 3. This setup allowed free motion for the hips, while constraining rotation of the knee joints, ankles, and feet.

In order to keep the motion of the cycle under tight control, I decided to use eight separate states to define the motion. This allowed me to keep the spline interpolation moving in the intended direction, and also enabled me to vary the number of tween frames to change the speed of the walk while maintaining the desired control.

The first state is what I call the "stretch" position. This is where one tripod of legs is at its foremost position, ready to move back and propel the body forward, while the other tripod is at its farthest back position, ready to move upward and forward to complete a half cycle. There are two of these stretches in a complete cycle, the first with one tripod forward, and the second with that tripod in the back position.

To make the first stretch, I selected the front and back foot on one side, and then the middle foot on the other. This formed the first tripod (the legs shown as blue in Figure 2). I then turned on the Constrain button and changed to Pick Objects mode to activate the IK*. Using interactive move, I deselected motion in the X- and Z-axes, and moved the feet forward in the Y-axis. Selecting the opposite tripod, I moved its feet backwards in the Y-axis. I went back to Pick Groups mode, picked the parent axis, and created the first state, naming it "walk1" (Figure 4).

In order to keep the movement of the legs at a steady pace, I made a "cycle gauge"—a simple measuring stick that used a grouped set of eight axes to represent the placement of one foot in each state. The first five axes were equidistant on the ground. The last three formed an upward arc, from the last axis on the ground, forward to the first. I used axes instead of points so they

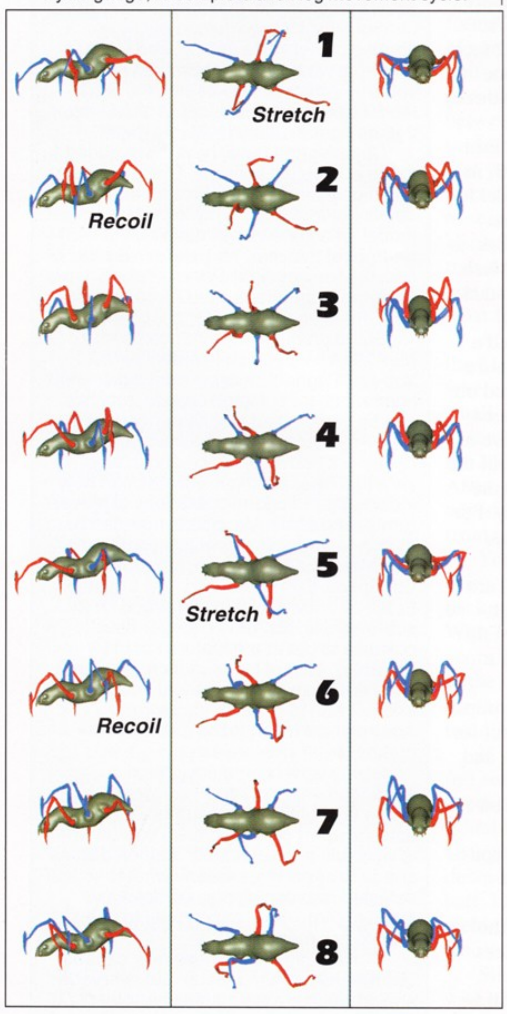
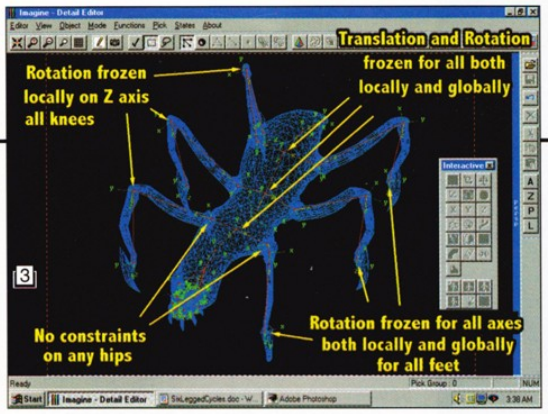


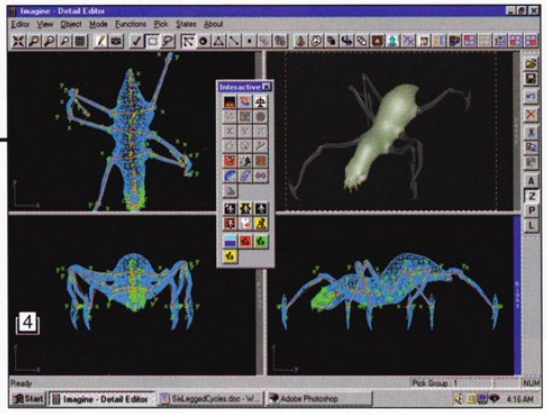
Fig. 2. The six legs form two tripods, each consisting of the front and back legs on one side and the middle leg of the other side. Here you can see states walk1 through walk8 (top to bottom) from the side and front, and from overhead. One tripod was moved first, through all eight states, using the cycle gauge on the side view. Then the other tripod was posed through the states, a half-cycle off from the red, using the same cycle gauge, to complete a full leg movement cycle.

*You can turn on Constrain at any time, but it will only constrain while in Pick Objects mode. With Pick Groups, objects move with forward kinematics, not inverse.

Until now, Impulse hasn't used the term, "inverse kinematics," but rather "Freeze" and "Constrain" (see "Freeze" in the Imagine manual index).

In the Detail Editor, you first States/Freeze the joints. Then you select which directions will be frozen. For example, if you want to rotate only on the Z-axis, check the freeze for the X- and Y-axes. The dialog, or requester, has freeze for local and global translation (the axis will not move back and forth in those directions), and local and global rotation. Once you have frozen the axes, you activate Constrain. Then, in Pick Objects mode, you select and move or rotate the axis. The rest will follow or stay still, depending on the selections you made in the freeze requester.





would remain visible in any mode (points disappear in Pick Objects or Groups modes).

I placed the first axis of the gauge on top of the foremost foot of the stretch state, then scaled the gauge until the last axis on the ground aligned with the matching foot on the opposite side (Figure 5). Individual foot movement didn't matter. I could use the gauge with only the front foot from each tripod, since the tripod's three feet move together.

To create the next state, "walk2," I selected the tripod with the feet forward and, using IK, slid its front foot back to the second axis on the gauge. Then, selecting the opposite tripod, I moved its first foot up to the first raised axis that starts the forward arc. (In Figure 5, this axis is labeled "walk6" for one tripod, but the other tripod hits it a half-cycle off, at the walk2 state.)

For the "walk3" state, the lead tripod's first foot was moved back one axis on the gauge, and the other tripod's front foot was moved up and forward to its next position. And again for "walk4," arriving at "walk5," which is the opposite stretch state.

Next the process was repeated, with the opposite tripods moving through the slide-back and arc-forward axis positions for states six through eight.

I used States/States/Animate in the Detail Editor to test the walk cycle, adding walk1 through walk8 with five frames each, and rendered. This animation was satisfactory, so I went on to create some overlapping motion.

To add the appearance of weight, I set the object state to walk2. When the motion is cycled, the feet of the blue tripod make contact with the ground at walk1. As the feet of this tripod move backwards to walk2, the body's weight pulls the body down, causing the legs to recoil. To achieve this look, I just moved the legs up on walk2. Only one tripod has to be adjusted, as the other tripod is off the ground at walk2. This "recoil" state reoccurs at walk6, where the red tripod bears the falling weight of the body.

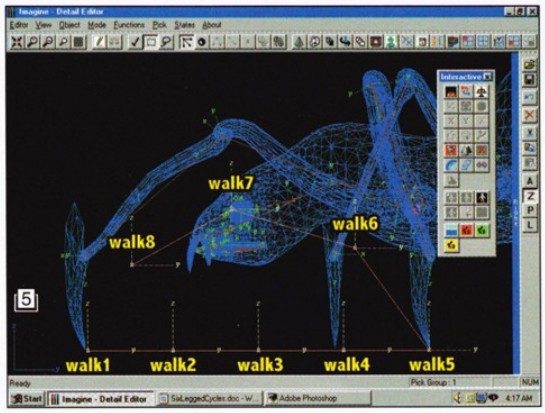
Next, I added some secondary motion for the body. To do this, I froze the feet in all directions for translation, and unfroze the joints in the center of the body. I rotated the center joints toward the motion of the legs, first from the top view, then from the front view, updating the respective states as I went along. This gave some convincing hip motion.

Something to keep in mind while adding the secondary motion is the sequence of events. In top view, walk1 and walk5 are the extreme positions for horizontal motion, while walk3 and walk7 are intermediate positions where the hips would be closest to being aligned straight across. Therefore, you rotate the hips farthest toward the direction of movement in walk1 and walk5, about half as much in walk2 and walk6, none in walk3 and walk7, and half again in walk4 and walk8 in the top view. The opposite is true for the front view, where walk3 and walk7 are the extreme positions for the vertical motion. This all keeps the movement at a steady pace (see Figure 6).

After adding these changes, I checked the cycle with a states animation. Everything moved well, but the walk looked too smooth and mechanical because I had moved the feet together as units of three. To fix this, I went back through each state and added some random foot and leg movement using Pick Groups mode and forward kinematics. I was careful not to disturb the feet in each state where there was contact with the ground.

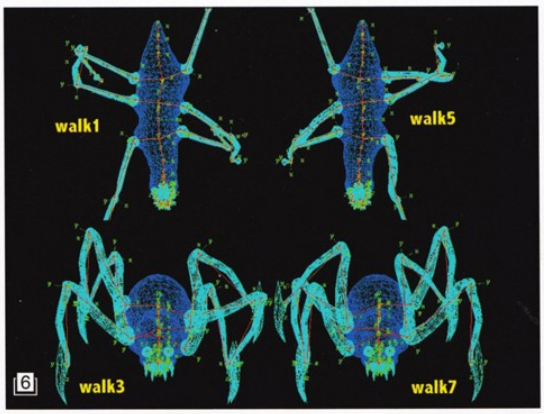
Now I had a convincing walk cycle for a six-legged creature. The next step is to load the object in the IFW Stage Editor and add the actor's states bars at five frames each. Once the first cycle is loaded, you just copy and paste the cycle down the remaining frames. Then use bluing* to position the object so the traveling feet remain on top of each other, and update the position bar for each state.

*Also known as "onion skinning," this shows previous frames underneath the current one, so you can see an object's earlier positions and move it accordingly.



See next page for more about this project.

R. Cory Collins *freelances part-time outside Macon, Georgia, and has been using Imagine since its earliest days on the Amiga. In* **3DA***#36, he wrote about facial animation and character design, and he is the author of the character animation chapters in Richard Foster's* I-Files *instructional CD for Imagine For Windows (***dspace.dial.pipex.com/rfostertg/ifiles.htm***). You can reach him at <rccollins@alltel.net>, or visit* **www.alltel.net/~rccollins**.



Above and right movie frames courtesy of Eagle Films, ©Copyright 1999 Eagle Films. Images below are test renders by R. Cory Collins.

Desktop Special Effects

The creature used in my "Six-Legged Walk Cycling" article on the preceding pages was created for *The Road*, a feature-length movie coming from producer/director Phil Cook and Eagle Films (**www.eaglefilms.com**). The script called for some interesting mutant creatures, and I was contacted for their design and animation. (Phil Cook himself is a longtime Imagine user who has incorporated it into many of his Eagle Films productions.)

Some sketches were passed back and forth for the look of the creatures, then I got down to work on creating them. I used Organica and Imagine for Windows, as well as Mike Clifton's freeware spline modeler for Win95/NT, SPatch (**www.cableone.net/alyson/spatch.html**). The creatures' bodies were made with Organica, the legs were created with sPatch, and they were all joined together, boned, textured, and animated within IFW 1.4.—*R.C.C.*

SPatch directly supports POV, DXF, VRML 2, and RenderMan, and third-party support is available for Moray and POVLab.

Textures & Light *Continued from page 36*

Even static objects can be dynamic if the color play is strong enough.

Textures and the color of the lights should agree with each other, and contribute to the overall color cast of the image. For instance, if you are using yellowish lighting, too much blue or cyan in a texture can look unnatural. You can make the color of the texture and lighting agree by eliminating any unnatural color casts in the texture before the lighting ever gets applied to it, or you can make sure the color cast is appropriate for the scene lighting.

For example, here we have a room with brick walls and a concrete floor (Figure 1). The textures are nice library textures, but they look a little bit like sterile carpets. They are tiled and have no relationship with each other or with their locations on the geometry. The cement also has a bluish cast to it, which doesn't really go well with the bricks and the overall warm tone of the light.

Textures that you use probably have a certain color balance. This is especially true of scans and scanned photographs, as all imaging equipment always introduces certain color biases into a raw image. Cameras, film, scanners, temperatures, and initial lighting conditions all influence the color balance of an image. Scanners, for instance, often give an image more blue and green than the original, but this is something you will have to experiment with for your own equipment.

Don't be afraid to use the color balance functions of your image editor for any texture. Play with the sliders, increase or decrease color values, and compare with the original.

Figure 2 shows the result of color correcting the textures in the example scene. I brought the cement more toward yellow and red, and desaturated the brick. The color balances are now more in character with each other, and with the tone of the lighting.

It is also helpful to build some of the lighting colors and effects into the textures themselves, if certain surfaces need to take on more or less of the lighting color. This can also be a great way to introduce some local reflected and distributed light effects without actually adding more light sources.

In Figure 3, I have added some local detail to the bricks and cement, including an appropriate transition between the two surfaces. This local detail is very important to texturing. In 3D imagery, too often a brick texture transitions to another texture in the middle of a brick. This is not how it would happen in the real world. The brick and cement do not meet at a perfect 90 degree angle with no regard for one another; there is a