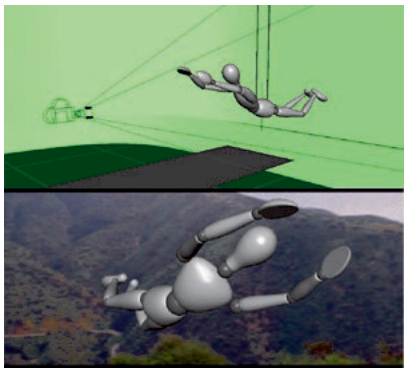
# ALIASWAVEFRONT MASTERCLASS
# San Diego July 2003

**Title of Seminar:** Previz, from the Director's Concept to the Stage

**Presenters:** Pepe Valencia / Dan Ziegler
Sony Pictures Imageworks, Culver City, CA. USA

**Overview**

Previsualization is a step beyond the storyboard, a way of visually planning a shot so that important decisions can be made before the camera rolls. This class will focus on using Maya as a previz tool, exploring both the aesthetic side of previz and the technical tools available in the software.



## I. Introduction
- What is Previz
- Why Previz

## II. Production Aspects
- Concept
- Different Pipelines Scenarios
- Production Aspects
- Sequence and Shot Structure

## III. Visual Aspects
- Framing and Composition 101
  - Where to place the camera
  - Negative Space
  - 180 Rule
  - Silhouette
  - Perspective vs. Symmetry
  - Rule of thirds
  - Camera Angles
- Shot Framing Convention
- Camera Moves

## IV. Technical Aspects
- CG Camera Setup
  - Basic camera setup
  - Complex camera setup
  - Camera setup using Mel scripts
  - Pan and Tile
- CG Camera Tools
  - Reticle and Masks Examples (1:85, 2:35)
  - Using Heads-Up Display for scene info
  - Preparing a scene for Stage
  - Rendering Scenes for use on set
  - Miscellaneous tools

# I. Introduction

## What is Previz

Previsualization or “previz” is a pre-production process that helps to analyze, visual and technically, the position, speed and movement of the camera and characters through a scene. This gives us an idea of how a shot will appear before it’s filmed.

## Why Previz

Previz is like a blueprint that helps to communicate the future shots to the whole production team. Let’s define how Previz helps different teams, for example:

- *Director* - has the ability to experiment with the scene and its elements. From camera angles, positions and speeds, to actors performance, illumination, set design, etc. He is able to communicate to others his final choice in a very specific way.

- *Actors* - grasp the director’s visual idea to accommodate into their “character”.

- *Producer* - has a clear plan of costs and timeline of the shots. Previz can save a lot of time and money.

- *Set Designers* - have a source to obtain exact specifications of elements that are needed, including set measurements, electrical and lighting equipment needs, camera positioning, etc. Previz helps them to match the CG elements with practical ones

- *Editors.* Can start to edit sequences using the previz shots, providing DP’s and directors a rough version of the whole picture.

# II. Production Aspects

## Concept

It is good from the beginning to clarify the needs of our production. This is where a *Digital Pipeline* becomes important. Each production is unique, so each time we should analyze the needs and design a pipeline to meet them.

## Different Pipeline Scenarios

First we need to analyze what kind of production it is: Is it full CG, Live Action or a mix of both?

- Digital Pipeline in a CG environment:
  Story - Storyboard – 2Dscan – Maya Previz – Final Comp

- Digital Pipeline in a CG environment plus audio and/or sound:
  Story - Storyboard - 2DScan - Sound - Avid - Maya Previz –
  Final Comp

- Digital Pipeline in CG – Life Action environment:
  Story - Storyboard - 2DScan – Sound - Avid - Maya Previz -
  Export Data to Stage - Stage Data to MM camera – Final Comp

## Previz Digital Pipeline

The following are different steps that we should observe in building our pipeline.

- ***Research Data***. The first step is to obtain as much information about the project as we can. From blueprints of the stage to actor's information, pictures, script, the storyboard, notes from the director, everything that can influence our previz.

- ***Prepare Data.*** In this stage we will configure our entire project in order to be ready to start.

    - *2D Scan of the Storyboard*. It is good idea to scan the storyboards and use compositing/editing software to create an animatic. As soon as we final our previz shots, we'll replace those storyboards with the new previz images.

    - *Audio or Sound,* if we are going to used audio or sound, prepare audio files to be loaded in Maya.

    - *Project Directory Structure,* have the previz models ready to go, as well as the cameras setups that we'll use in our shots, e.g.

/*Model.* Low res Models of characters, props and environments.
/model/characters/
/model/environment/
/model/props/
/Camera/ Camera Setup that will use in our Previz Project.
/Storyboard/ with all the 2D images already scan
/Audio

- *Previs Tools,* have all the tools prepared, including mel scripts and a previz shelf.

- ***Previz.*** Now that everything is ready, it is time to determine the timing for the characters and the camera and transfer this information to the next level, the stage.

  - *Creating a Project.* Generate the scene using the naming conventions. More info below.

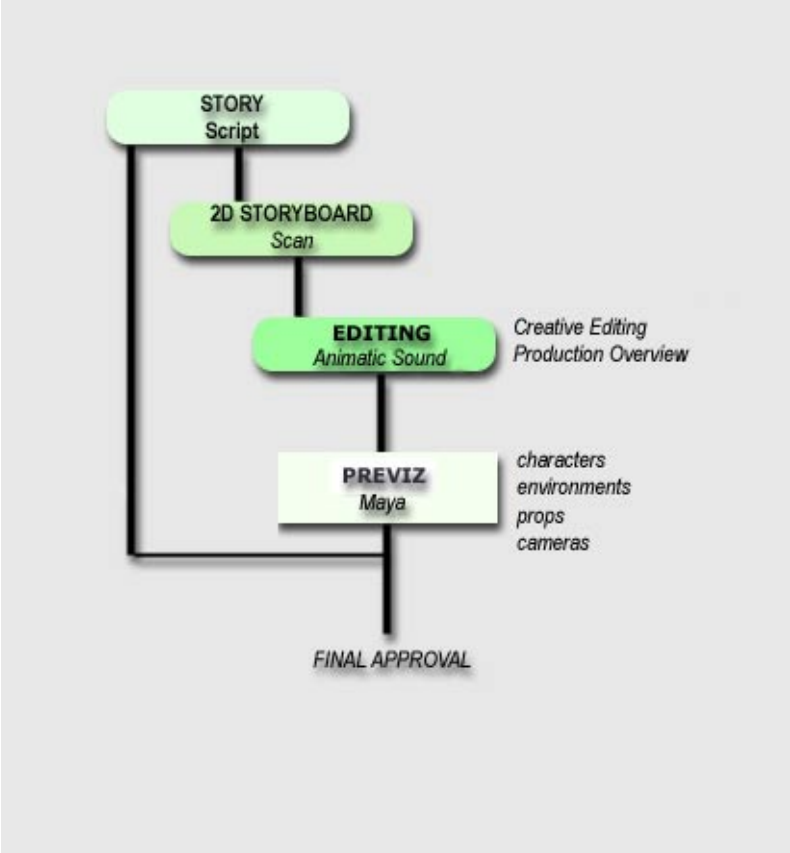
  - ***I****mporting Models and Cameras.* Import models, environment and props. Rough Layout may use low-rez geometry. We should be flexible and have the ability to create as many cameras as we need to create the scene, at the end just one "camera_final" will be the camera for each shot.

  - *Importing sound*/*audio* into the scene.

  - *2D animatic* will provided video and audio from the cut for each shot in the Avid.

  - *Camera and Character Blocking*. Blocking of the scenes, rough animation of the characters and a final animation of the camera.

  - *Export Camera To Stage*, final Step.




*Previs Pipeline*

## Sequence and Shot structure

A sequence is a series of shots that make up an action. A shot is a continous take in that sequence, and takes are variations on each shot. You need to plan how to create these shots as maya files. There are different options for setting up a shot, for instance you can create an individual animation per shot and add a camera to it, or create a single master animation for the entire sequence and place all the cameras in that scene. Here are some different approaches:

- **Cinematography - Editing Creation**
  Once you've created the action sequence, place all the cameras for the shots and render each camera. Later on, the director and editor can use the virtual "footage" to edit the sequence. Each camera needs to be rendered for the whole sequence length.

  *Pros:* The director and editor have the ability to cut and define the sequence.
  *Cons:* The time to render the whole sequence for each camera.

  Example:

/sig03/shot01maya/project/scenes/shot_master_previz_v1.mb
(100 frames).
cam_shot01 1-100
cam_shot02 1-100
cam_shot03 1-100

- **Non Linear Creation**
  Create the entire action sequence and at the same time all the cameras for the entire sequence. Once roughly approved, break down into different shots and finesse each shot.

  *Pros:* you work entirely in one scene, any change in animation affects all the cameras at the same time.
  *Cons:* sometimes you want to "cheat" the time or the animation to work for a specific camera.

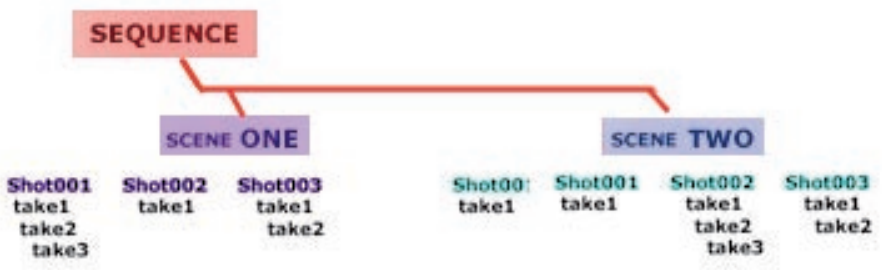
  Example:
  /sig03/shot01maya/project/scenes/shot_master_previz_v1.mb
  cam_shot01 1-35
  cam_shot02 36-75
  cam_shot03 76-100

- **Linear Creation**
  This is like straight-forward animation, you create the action specifically for each shot. If later on there is any change, you version up the file.
  *Pros*: You have more control of the animation and the camera, working independently of the other cameras.
  *Cons*: If you need to make significant changes to the whole action, you need to go shot by shot and make the adjustments.

  Example:
  /sig03/shot01/maya/project/scenes/shot01_previz_v1.mb 1-35
  /sig03/shot02/maya/project/scenes/shot02_previz_v1.mb 36-75
  /sig03/shot03/maya/project/scenes/shot03_previz_v1.mb 76-100

## Tips for Cameras, Animation and Versioning

- Name your cameras and files consistently using naming conventions.
- For changes to existing cameras, adjust camera and version up. You never know if you'll go back to use the old camera again.
- For changes to existing animation, adjust animation, make sure it works for all cameras and version up.
- If you change an animation and it does not work for all cameras, and you now need different animation for a specific camera, that animation and camera must be made into a new shot.
- Any time you are asked for additional animation that does not fit into an existing shot, create a new shot.
- If you work in a group, and your scene is going to another animator, try to keep it clean. It's a good idea to write down what you did.




*Previz shot structure*

# III. Visual Aspects

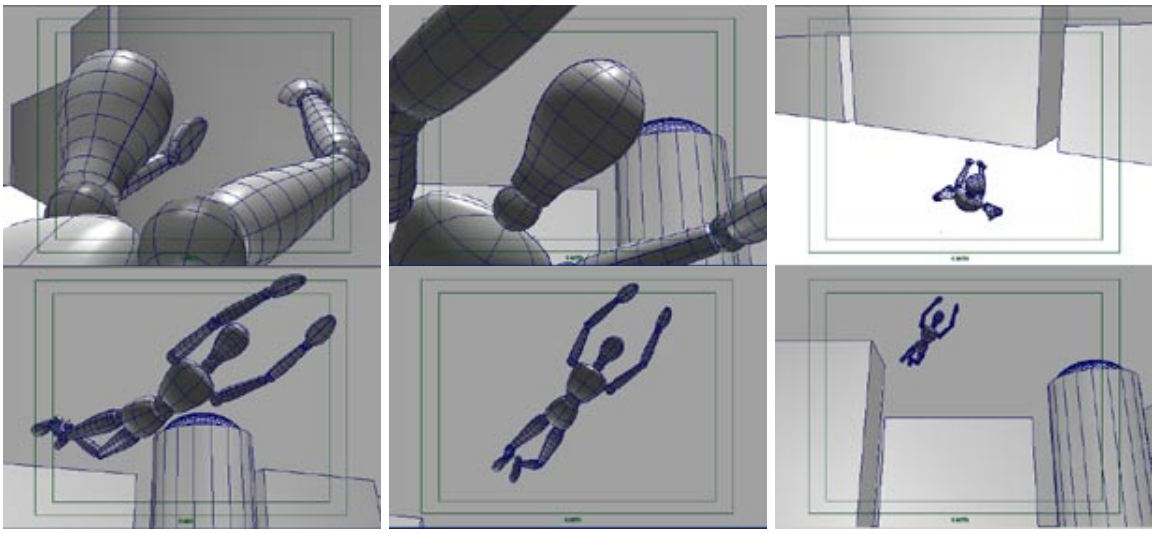
People read pictures as they do with words. Pictures, like sentences, have their own grammar. In order to tell a story visually we need to know the principles or elements of the shot.

## Framing and Composition 101

The ***framing*** is the selection of the view. To arrange the objects and/or characters in front of the camera is the ***composition***. We should show the audience what they need to see.

A **shot** is the smallest possible unit when shooting, and contains visual information for the audience to follow the scene. A **Scene** is a group of shots placed in a **sequence**. In every shot we frame a view using the camera, choosing a portion of the view to provide the audience with information about the character or the subject.

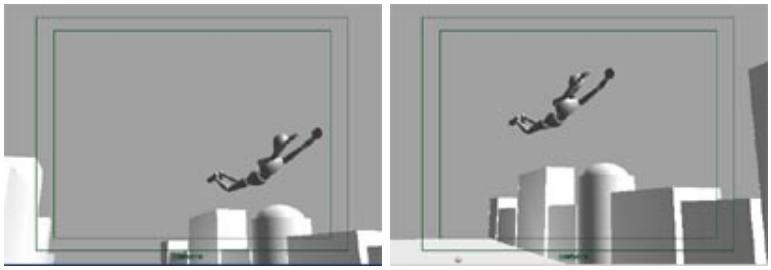
**Where should I locate the camera?**
In the best place to read the action. Try to generate as many ideas as possible, because previz is the time to play around with the camera and create different options, selecting the best ones.



*Different camera placements*

## Negative – Empty Space

It is important to strike a balance between positive (filled) and negative (empty) space. Try not to leave a large part of the frame empty while the rest is cluttered.

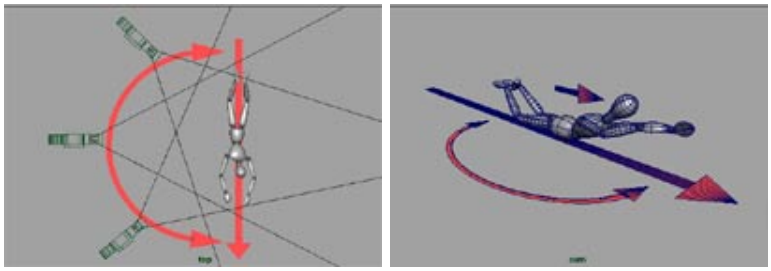


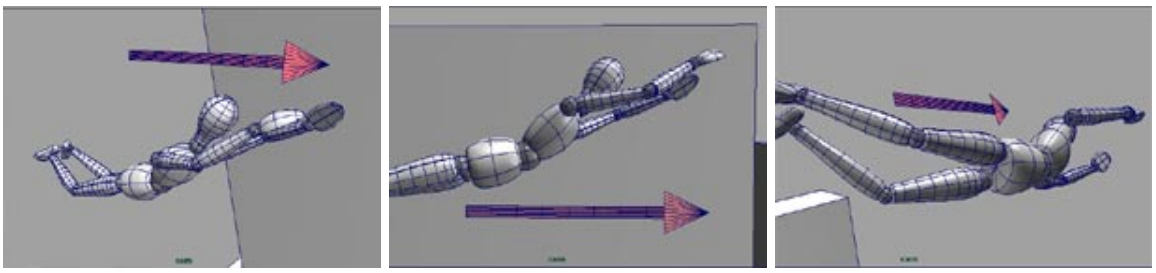
*Negative Space*

## 180 rule

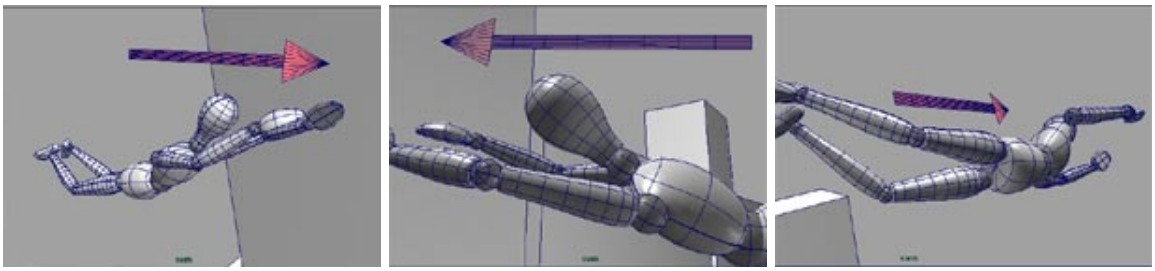
The idea is to establish a line of action between the characters or the character direction. Once defined, we place our cameras on one or the other side of this line.

To understand this principle, imagine a scene of two armies charging at each other, one moving from screen-left to screen-right, the other from right to left. If the camera is rotated 180 degrees between shots, the army that was charging to the right is now going left, and vice versa. This can confuse the audience as to who's who, and which direction they're going.





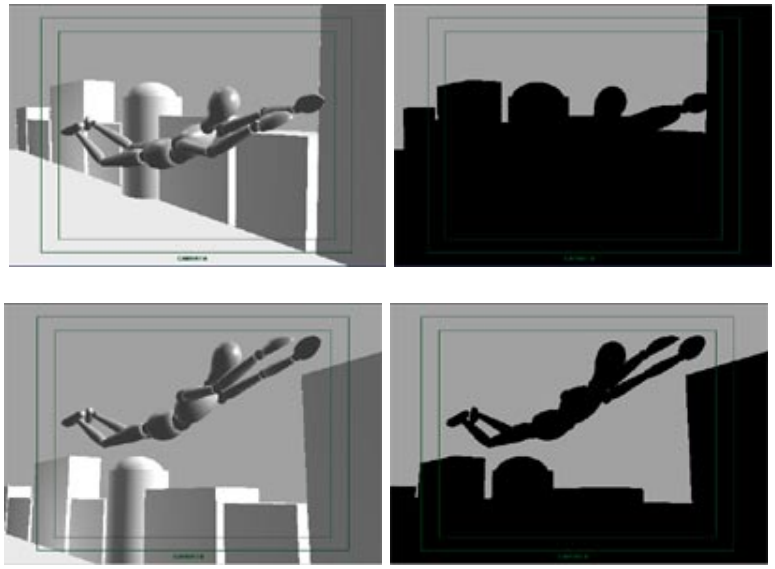
*good direction*



*wrong direction*

## Silhouette

Try to make the character read as clearly as possible. Using the silhouette of our character as traditional animators used to is a good idea.



*Silhouette*

*See previs shelf button: "Silhouette"*

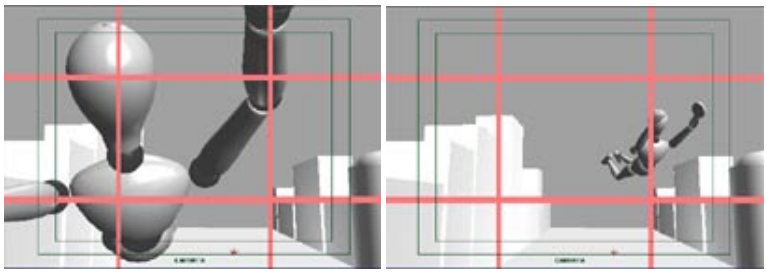
### Perspective vs. Parallelism

It is good to look for perspective cues, try to avoid flat compositions by looking for angles instead of straight lines. Avoid placing the camera completely parallel to the subject.

*flat composition vs. perspective*

### Rule of thirds

The rule of thirds suggests that if you divide the screen into thirds, your subject should be placed at a point where the lines intersect.

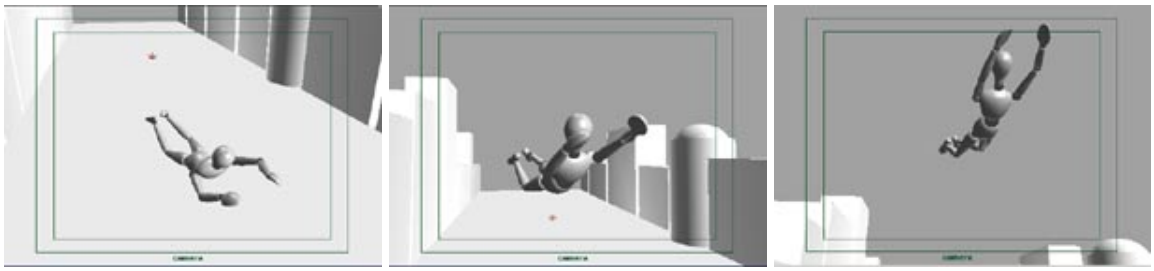


*example of Rule of Thirds*

*See previs shelf button: "thirds overlay"*

### Camera Angles

The camera angle helps to determine the point of view of the audience. This is very important since they have seen a lot of TV and film, and have been conditioned to interpret the cameras "eye level" as containing meaning.



*high-angle, eye level and and low-angle*

**How to move the camera**

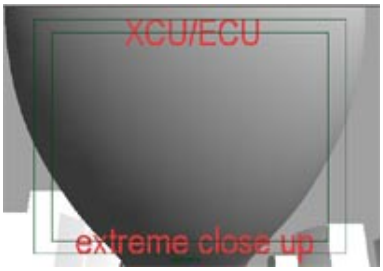
- The camera has to move for a reason. No only characters creates the action, camera moves have importance too.
- Avoid motion that draws attention to the camera rather than to the subject.
- Give the subject space to move.
- Give the viewer time to look at something before moving on.
- Move at a speed such that the viewer can actually follow the subject.
- Don't anticipate the movement of the subject with the camera/

**Transitions between Shots, Continuity and Overlapping**

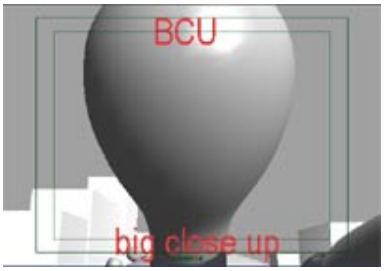
- The action should cut in a clear way, try to stay away from confusing the audience with impossible camera angles and positions.
- Whenever possible, cut within an action.
- We need to look for continuity between shots.
- Expanding and compressing time: Overlapping shots show the drama of a fast action by expanding time. Less interesting actions can be compressed into quick shots.
- For overlapping shots, match the speed and continuity of the action.
- Check that the line of action works between shots.
- Use consistent head room for characters.

**Shot Framing Convention**

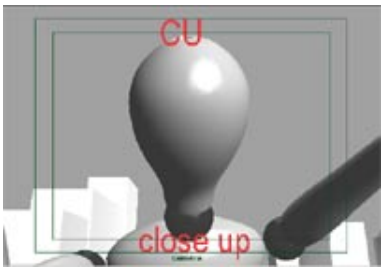
It is important for different teams to use the same shot framing name convention.



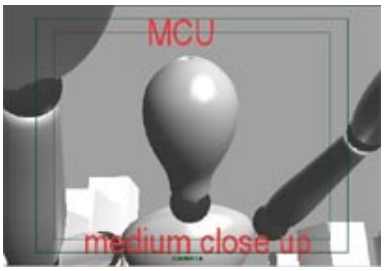

*XCU/ECU Extreme Close Up*



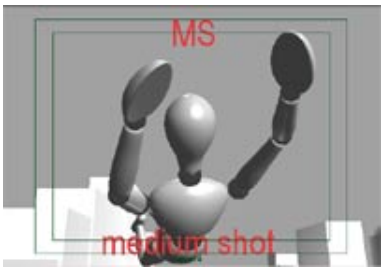

*BCU Big close up*




*CU Close Up*




*MCU Medium Close Up*



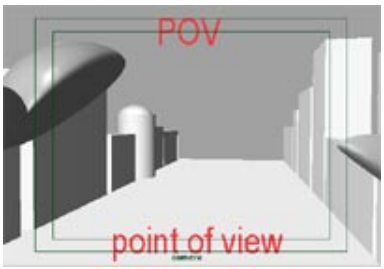
MS Medium Shot

*MLS/MWS Medium Long/Wide Shot*

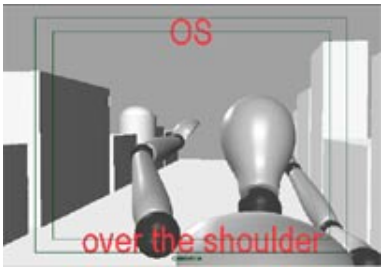
*LS/WS Long/Wide Shot*

*VLS Very Long shot*

*XWS/EWS Extreme long shot / Wide shot*



*POV Point of View*

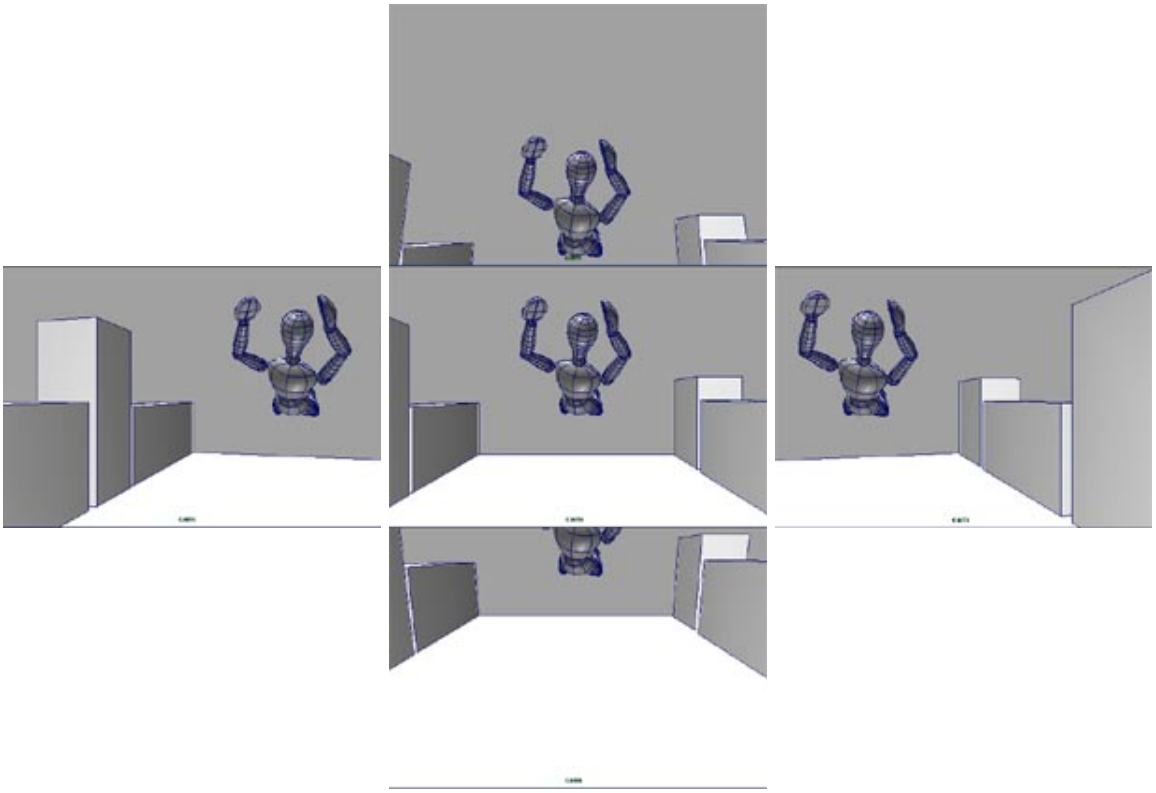


*OS Over the shoulder.*

It is a good idea to use the safe title or safe area, in order to know when you exceed the limits when framing.

## Camera moves

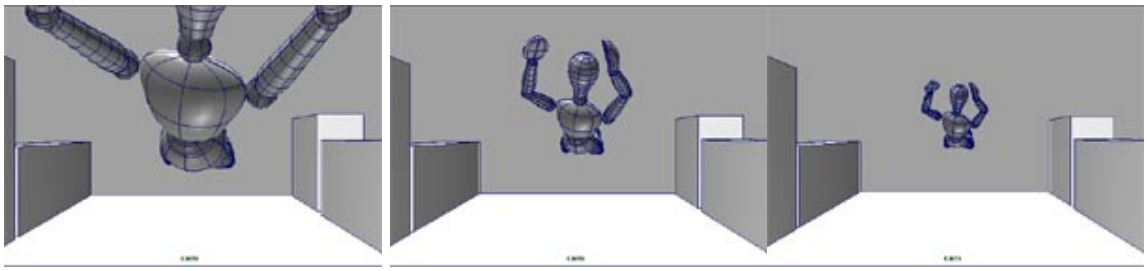
***Tilt (rotX) Pan (rotY)***
Two of the most common camera moves in filmmaking are pan and tilt.
The camera rotates in a horizontal (panning) or vertical (tilting) plane.
Uses: Useful to move the camera from one subject to other, to reveal the space around the subject.to follow a moving object or character, or to show more than can fit into a single frame, such as panning across a landscape. It is also used as a transition between one camera position and another.
Tilting or panning too fast causes an effect known as strobing.



*(Pan right = +rotY,  Pan left = -rotY)(Tilt up = +rotX ,  Tilt down = -rotX)*

***Dolly (transZ)***

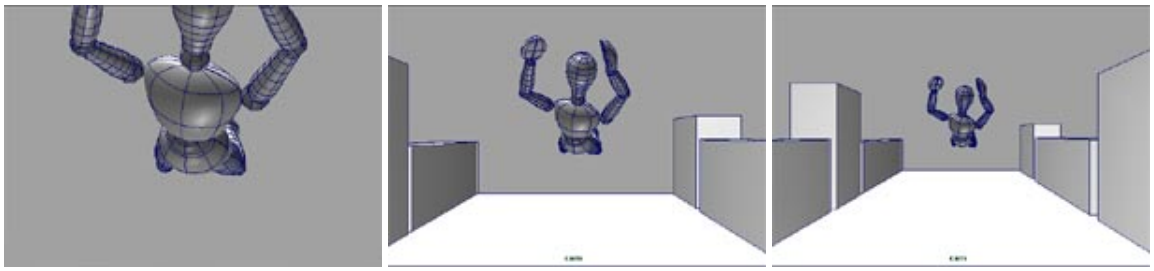
A dolly moves the camera in and out of a scene. Our actual camera translates away or towards our subject.



*( dolly in = +transZ;  dolly out = -transZ )*

***Zoom (focal length)***

Where we change our perspective of the subject by changing the lens, without actually moving the camera.



*(zoom in = + focal length,  zoom out -focal length)*

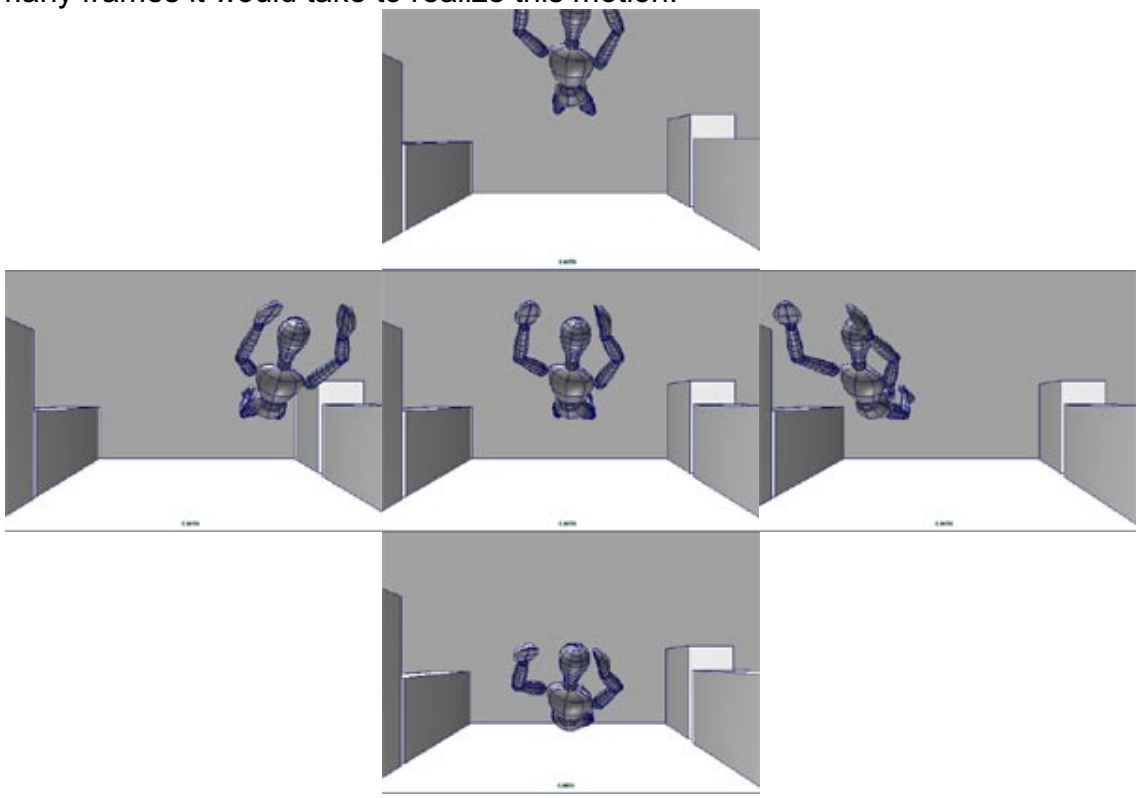
**Dolly vs. Zoom**

They have similar effect but a dolly physically moves the camera, while a zoom just changes the focal length.

***Track (transX,Y)***

A tracking shot is a movement perpendicular to the camera lens axis. The key to these shots is to have realistic motion.

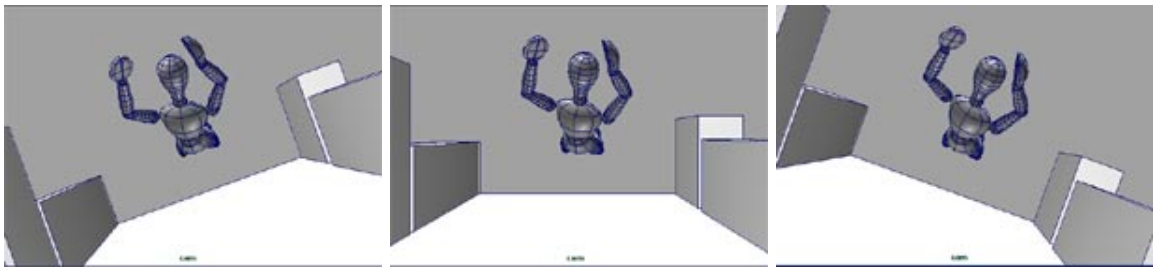
The motion can be judged by looking at how fast humans move and then how many frames it would take to realize this motion.



*track right (+transX) track left (-transX) truck-crane-boom up (+trackY) truck-crane-boom down (-transY)*

***Roll (rotZ)***

When the camera rotates around its "z" axes"



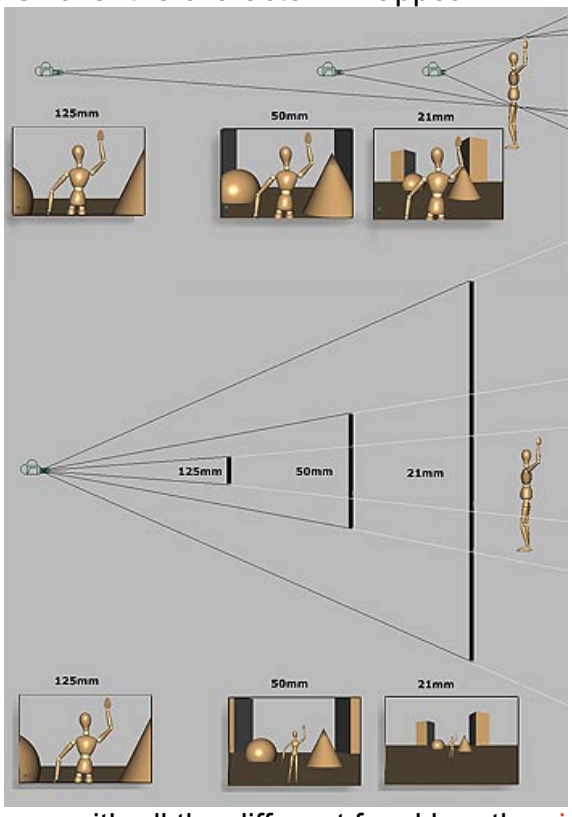
rolling right (r+rotZ) rolling left (-rotZ)

**Combination of moves**

Combining different types of camera moves can have interesting results. For instance, the “Vertigo” shot is a combination of dolly and zoom which gives the impression of zooming the background while keeping the character at the same scale.

**Focal Length**

Measurement of the ratio between the diameter of the lens and the distance from its optical center to the focal plane, usually given in millimeters. The longer the measurement the greater the enlargement of the character; the shorter the measurement the smaller the character will appear.




Download the scene with all the different focal lengths sig03_lenses1.mb
Download the scene with all the different focal lengths sig03_lenses2.mb

# III. Technical Aspects

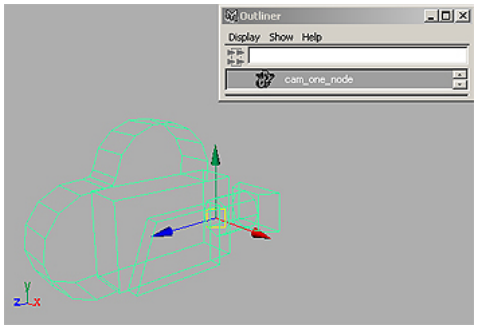
Here are some examples of tools available in Maya to aid the previs process. CG camera setup covers different techniques for effective simulation of a real camera. Pan-and-tile gives you the flexibility to create virtual camera moves using real photography. Reticles and masks ensure accurate framing, and heads-up display information is a visual way of making shot data available on set. Preparing a scene for the stage is the process of taking a freely animated scene and turning it into something that can be shot on a real set. This section will also address previsualization of motion control camera moves. Some useful previs mel scripts (included on the cdrom) are listed at the end of this section.

## CG Camera setup

When setting up a CG Camera, we should think about how the camera works in the real world. Let's talk about setting up cameras, from default cameras to more complex cameras using hierarchies, constraints or mel scripts.

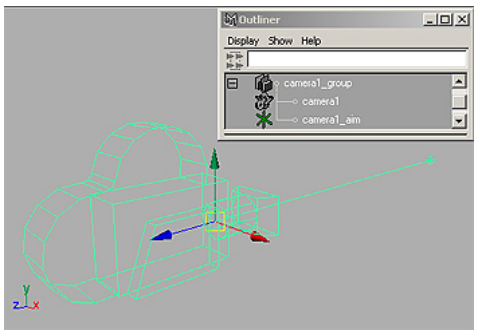
Maya provide three default cameras:

**One node camera**. The most basic one, used for static camera shots or little camera movement.



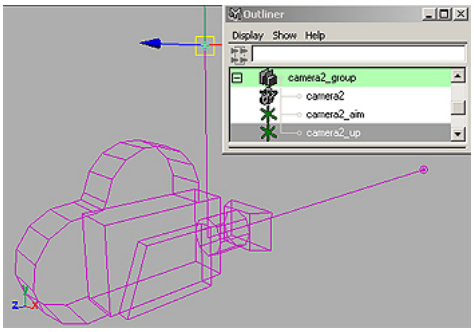

*one node camera*

**Two node camera**, adds a target and the camera is aim-constrain to it. Is ideal when linking the target to animated objects.




*two node camera*

**Three node camera**, adds the up vector to control more the rotation of the camera.




*three node camera*

## Camera Setup Using Constraints

- We can set up more elaborate cameras using constraints.
- Perfect setup to deal with cameras following objects.
- Cameras animating around the objects keeping equidistant from them.
- Flexibility in the moves thank to the offsetting of the camera and the object.

Step-by-Step
sig03_cam_constrain.mb

1. Create 4 locators, and rename them as:
   "main_group" , "char_group" , "cam_rot" and "cam_trans".
2. Create a one node camera.
3. Import our model geometry example: *sig03_char_man.mb*

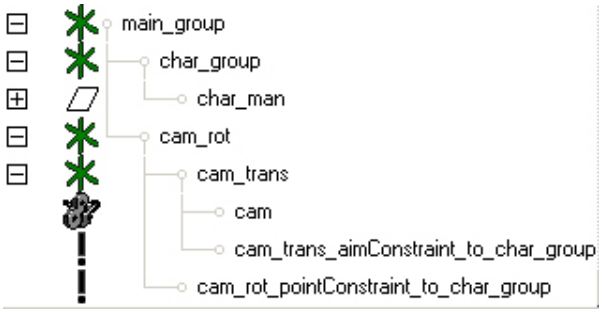
Now that we have all the objects, lets create the hierarchy.

4. Parent the camera to the cam_trans locator
5. Parent the cam_trans locator group to the Cam_rot locator
6. Parent the char_man or the sphere geometry to the Char_group locator
7. Parent the char_group to the main_group locator
8. Parent the cam_rot to the main_group locator

Now we are ready for the constraints.

9. Select the char_group, Shift-click and select the cam_trans, Constrain>Aim constrain. (Change the axes to constrain to 0, 0, -1)
10. Select the char_group, Shift-click and select the cam_rot, Constrain> point.

Open your outliner and compare it with the image below.




*camera constraint example*

Now let's see what we got:

Main_group, that animates all.
Cam_rot, rotates the camera around the character
Cam_trans, translates the camera around the character
Camera can be animated itself, acting as an offset
Char_man, can be animated itself, acting as an offset.

Download sig03_cam_constrain_sce.mb, as an example of animating the camera.
Download the mov sig03_cam_constrain_sce.mov

## Camera Setup for Cinematography Camera

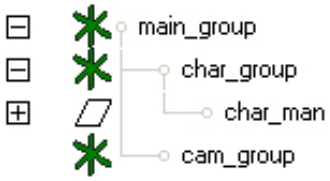
- Camera created by a hierarchy of nodes, that has all the axes of movement of a real camera.
- Gets rid of the gimbal lock errors thank to a hierarchy of locators.
- We will create a node with extra attributes, to control the camera through only one node.
- Use of an object or character as our "look at" node.

sig03_cam_hierearchy.mb

*Step by Step:*

- Create 3 locators, and rename them as: “main_group” , “char_group” , “cam_group”.
- Import our geometry example: *sig03_char_man.mb*.
- Parent char_man to char_group.
- Parent char_group and cam_group to main_group.

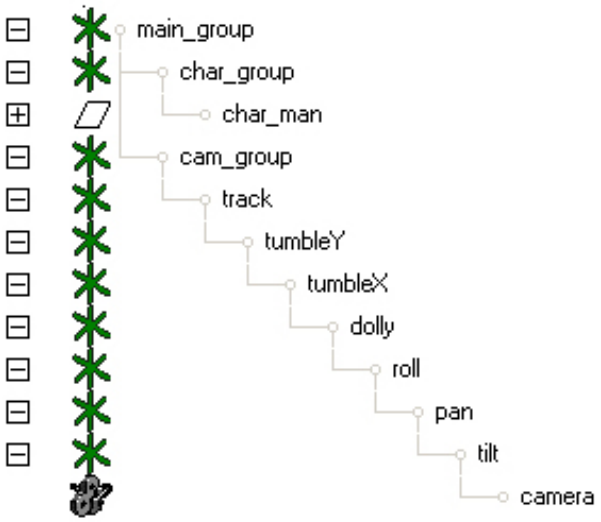
Open your outliner and compare it with the image below.




Now lets create all the locators for the camera moves.

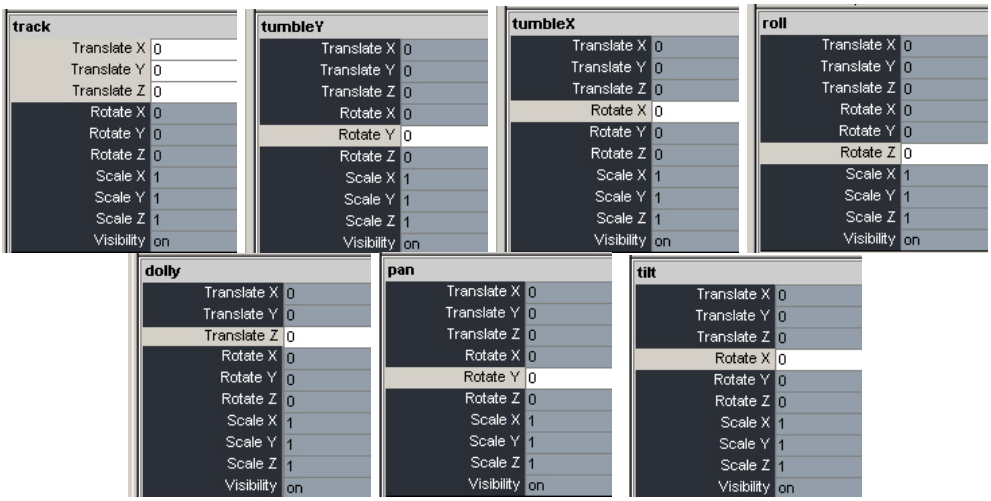
- Create 7 locators and rename them as : “track”, “tumbleY”, “tumbleX”, “dolly”, “roll”, “pan” and “tilt”.
- Create a one-node camera.
- Parent the camera to the tilt locator, then the tilt to the pan, then the pan to the roll, roll to the dolly, dolly to the tumbleX, tumbleX to the tumbleY, tumble Y to the track, and finally track to the cam_group.

Open your outliner and compare it with the image below.




It is a good idea to lock all the attributes that we don't need, and keep the ones that we'll use.

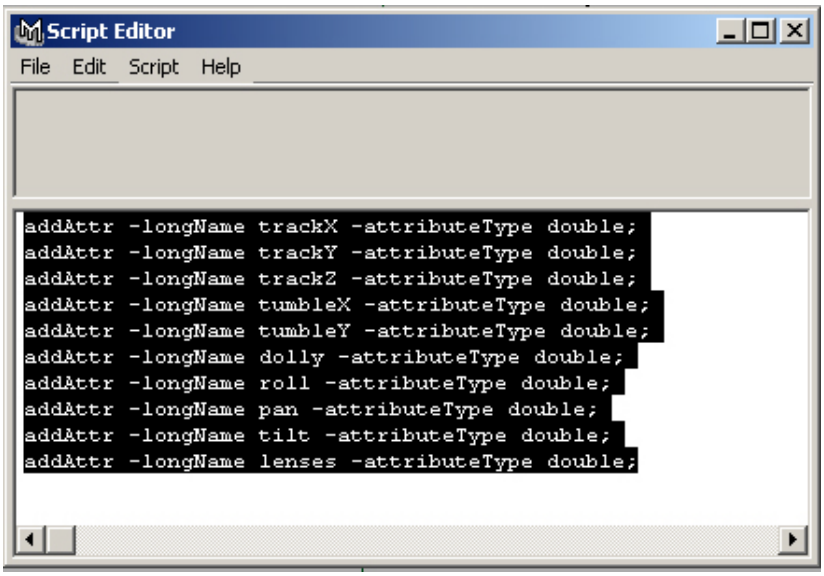
- Select the tilt node and in the Channel box and lock everything except the rotationX.
- Let's do the same thing for the other nodes:
  - Pan: lock everything except the rotationY.
  - Roll: lock everything except the rotationZ.
  - Dolly: lock everything except translationZ.
  - TumbleX: lock everything except rotationX.
  - TumbleY: lock everything except rotationY.
  - Track: lock everything except translation X,Y and Z.




*locked channels*

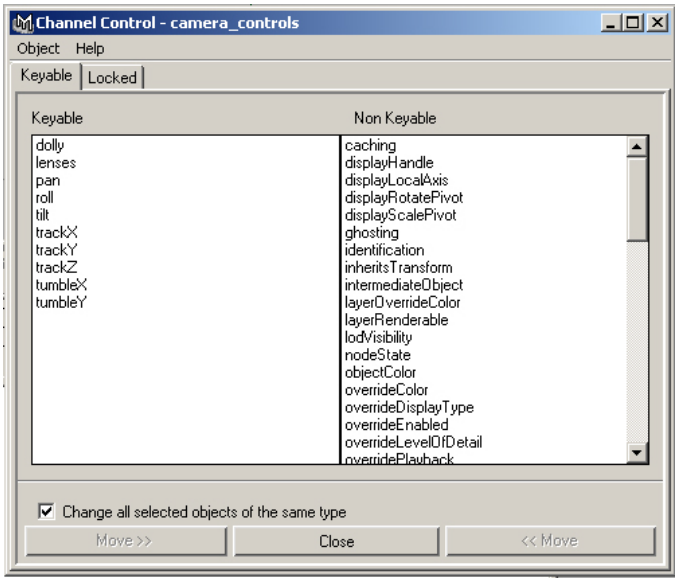
- Create an empty group and rename it "camera_controls".

- In the script editor add new attributes: trackX,trackY,trackZ,tumbleY,tumbleX,dolly,roll,pan,tilt and lenses.



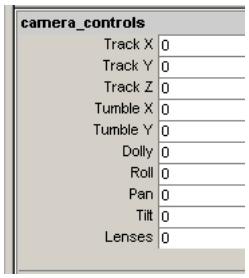

*add custom attributes*

- In the Channel Control Editor, make the new attributes keyable and the standard ones unkeyable.

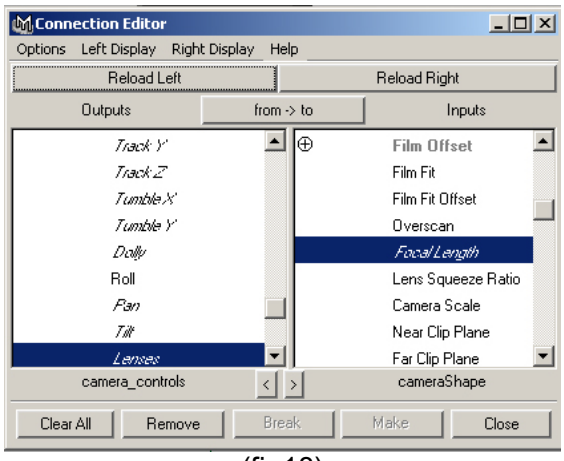



*make attributes keyeable*

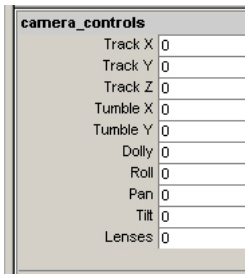
…this way we'll see only the new attributes in the channel box.




- Connect the new attributes we created in the camera_controls node to the ones from the camera group.
- Open the connection editor, reload left the "camera_controls" node.
- Reload right track node and connect camera_controls.trackX to track.translateX.
- Connect camera_controls.trackY to track.translateY.
- Connect camera_controls.trackZ to track.translateZ.
- Reload right tumbleX node from the outliner and connect camera_controls.tumbleX to tumbleX.rotateX.
- Reload right tumbleY node from the outliner and connect camera_controls.tumbleY to tumbleY.rotateY.
- Reload right dolly node and connect camera_controls.dolly to dolly.translateZ.
- Reload right roll node and connect camera_controls.roll to roll.rotateZ.
- Reload right pan node and connect camera_controls.pan to pan.rotateY.
- Reload right tilt node and connect camera_controls.tilt to tilt.rotateX.
- Reload right cameraShape node and connect camera_controls.lenses to cameraShape.focalLength.




(fig18)



(fig19)

Download sig03_Cam_hierarchy_sce.mb, as an example of animating the camera.
Download the movie *sig03_cam_hierarchy_sce.mov*

**Camera Setup Using Mel Scripts**

Sometimes we need to keep an eye on the face of a character or some other detail. Using MEL Scripts, we can generate cameras to be attached to the character, so that way we can see the details in any moment.

*See shelf button "create object camera"*

Having a zoomed-in view of your scene in addition to the regular view can be useful.

*See shelf button "create zoom camera"*

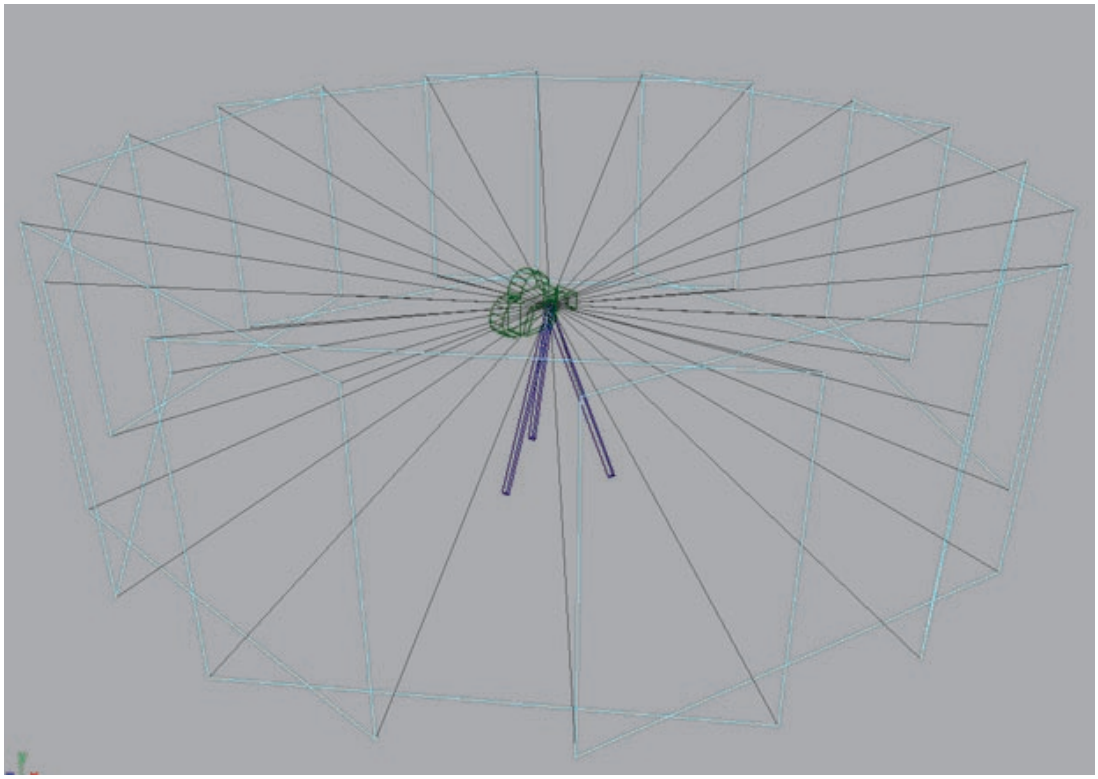
When you work with a scene at a real-world scale, our camera icon looks very small. Scaling the camera can sometimes cause problems when rendering, also the camera icon is not visible in a software render. A MEL script can be used to create "dummy" geometry that is constrained to the camera, which can be scaled and rendered.

*See shelf button "create camera dummy"*

## Pan-and-tile

“Pan and Tile” is a technique that’s becoming popular for creating virtual camera moves using real photography. By stitching multiple images together you can create an image that’s much larger than a normal frame, and you can the pan and tilt across this image in the computer. This also removes the need for matchmoving the camera, since the move is created in the computer.

The method for acquiring the images is simple. A series of photographs is taken from a single location, with the camera being panned slightly between each shot.



*Pan-and-tile camera*

In this example the camera is panned 36° per shot, resulting in 10 shots to cover 360°. For the stitching to work correctly there needs to be some overlap between each position.

## Assembling the images in Maya

1. Create and position a tile to fill the camera view

- Create a camera, make sure the *filmback* and *focal length* are the same as the camera that was used to shoot the tiles.
- Create a polygon plane named “tile”.
- Set the *width* attribute of the tile shape node to the same value as the camera’s *Film Aspect Ratio*.
- Parent the tile to the camera, and make sure it’s transforms are at 0.
- Rotate the tile 90 degrees on the X axis so it’s perpendicular to the camera.
- Move the tile away from the camera along its local Z axis.
- Looking through the camera, choose the panel menu *View>Camera Settings*/*Film Gate*.
- Scale the tile until it matches the film gate of the camera.

2. Create a material for the tile

- Create a new Surface Shader material called “tileSG1”, and assign the first tile image as a file texture.
- Set the *hardware texture channel* to color and the *hardware texture quality* to *highest*.
- Assign the shader to the tile.

3. Position the rest of the tiles

- Duplicate this tile and unparent it from the camera.
- Duplicate tileSG1 and assign the next tile image to the new file texture node.
- Assign tileSG2 to the tile still parented under the camera.
- Rotate the camera until tile2 lines up with tile1. If you know the exact rotations that each tile was shot with, you can simply plug that number into the camera and it should line up. If you don’t know the exact values, rotate the camera until they line up by eye.
- Continue this process until all the tiles have been created.
- Group the tiles together under one node. This group node can be scaled to move the tiles closer and further from camera without changing the perspective.

### Assembling the images with stitcher software

While assembling the images in Maya does work, there are often seams at the edges which are difficult to get rid of. Using special stitching software you can combine images into one panoramic image, and most seams are removed. I've used a program called PhotoStich to assemble these 12 images into one panorama:

*Panoramic image*

One you have the panoramic image, it's simple to use it in Maya:

- Create a nurbs cylinder
- Divide one of the tile images' height by its width.
- Set the *height ratio* of the cylinder to that value.
- Set the *radius* to how far you want the cylinder to be from the camera
- Create a *surface shader* and map the panorama as a file texture.

### Rules of pan-and-tile

- All tiles must be shot from same point in space, with only nodal rotations of the camera
- When viewed through a cg camera, the camera and the tiles cannot translate relative to each other. An easy way to prevent this is to group the tiles together and point-constrain the group to the cg camera.
- The only virtual "camera moves" allowed are nodal rotations and zoom (focal length).
- Tiles should be shot with a wider focal length than the cg camera to ensure that the whole frame is covered.
- Tiles do not have to be shot in a cylindrical configuration, a whole sphere of tiles can be shot for a complete environment.

### Uses of pan-and-tile

Pan-and-tile can be used to previz camera moves that are shot "locked-off" (unmoving) or nodally rotating. It can also be used to previz a moving camera for objects that are far away, since they are not affected greatly by camera movement. For instance, if you were previsualizing a shot where the camera orbited around two characters who were standing on a mountain-top, you could create a pan-and-tile backdrop for the distant mountains. The advantage of this over cg models is that the backdrop will look much more realistic. Pan-and-tile is more than just a previz technique, it is often used as an element in final composites. Using a motion picture camera, moving footage is shot using the

same rule of regular nodal rotations. Stitching can be more complex when dealing with moving images, but the principal is the same.

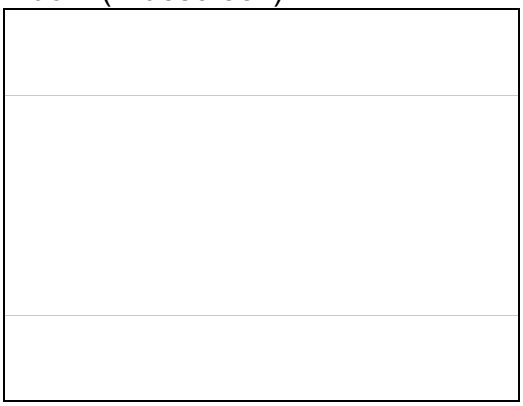
Quicktime example: *pan_and_tile_fly.mov*

## Reticles and Masks

Most film is shot with a different aspect ratio than the final print (the aspect ratio is the ratio of screen width to height). For instance, even though movies are usually presented in a wide-screen format (e.g. 1.85:1 or 2.35:1), they are often shot “full aperture” (4:3). To change the aspect ratio from full aperture to widescreen, the image is cropped.

When previsualising a shot that will eventually be cropped, it is important to show the crop area, and to frame the action within this area. A way to do this in Maya is to use an imageplane that it attached to the camera, with the crop area defined as a line (reticle) or black rectangles (mask).

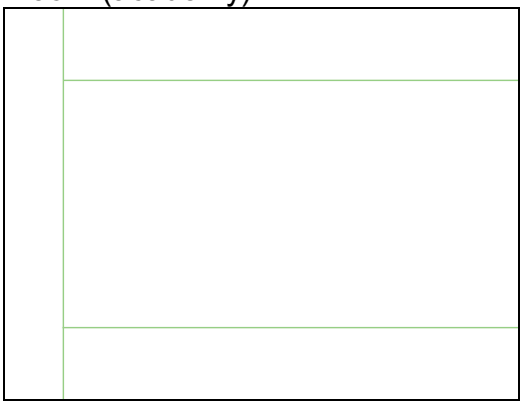
2.35:1 (widescreen)



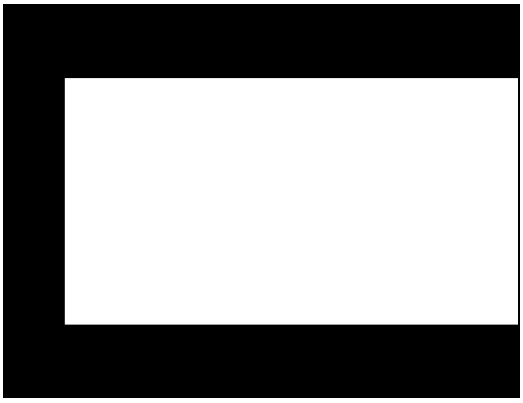
*235 Reticle*

*235 Mask*

1.85:1 (academy)



*185 Reticle*

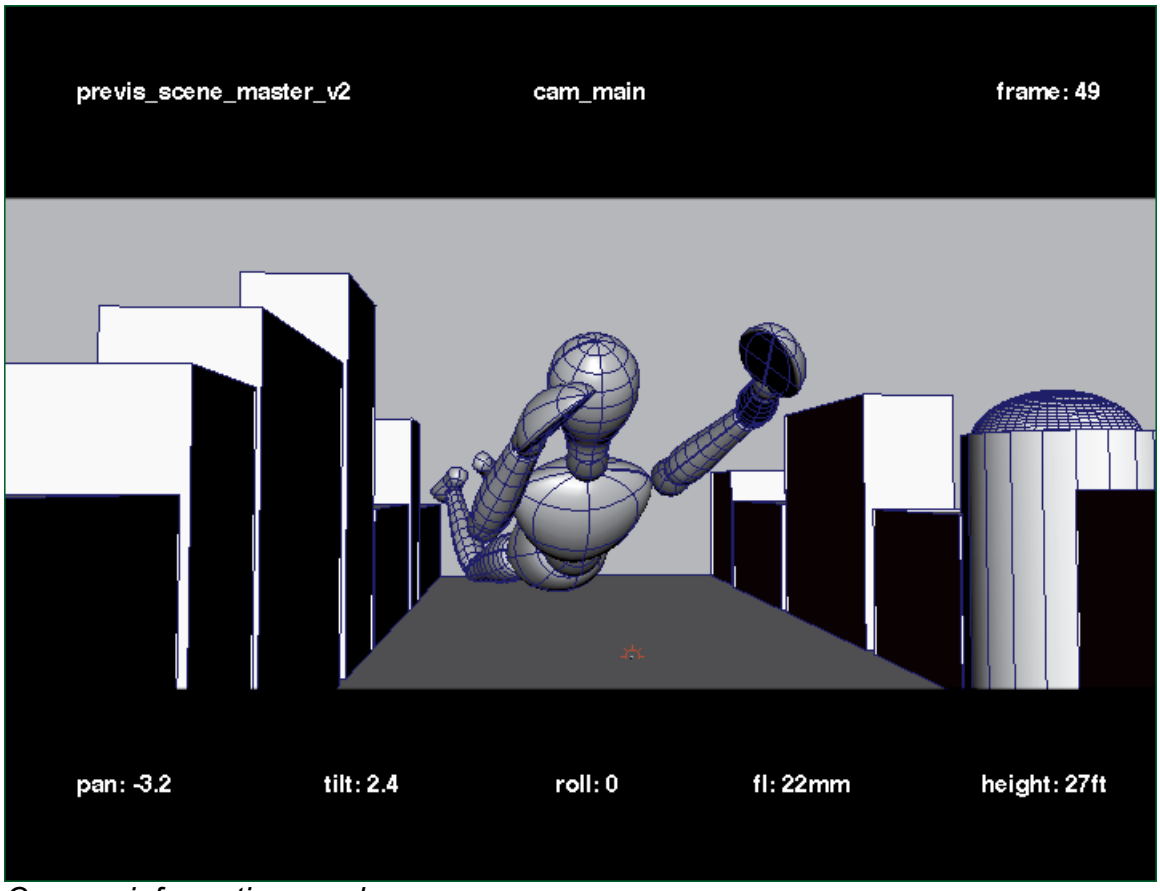


*185 Mask*

## Information Overlay In Playblasts

When you do previz in a 3d software environment like Maya, you have access to a lot of “real world” information about your scene. Assuming you’ve accurately modeled your set geometry and camera, you can find out useful values such as the camera focal length, the height from the ground, camera rotation, etc. The question is, how do you provide that information to the director or cinematographer who won’t be working with the 3d software directly? You can print out the values on paper, which works well when the camera is static, but what about when it’s moving? You need a visual way to display the information on a per-frame basis.

Using Maya’s built in *heads-up display* tool, you can overlay text on top of your playblasts, and have that text update dynamically.




*Camera information overlay*

This shows an example of information overlay, displaying:

- scene name
- camera name
- frame number
- camera rotation values (pan, tilt and roll)
- focal length
- camera height

A 2.35:1 mask has also been applied to the camera.

*see shelf button "previs HUD"*
*see shelf button "235 mask"*

A playblast of this scene can be copied to a video tape or dvd to be viewed on set. This provides a quick visual reference of the camera move.

Other useful information that could be put in an overlay include notes from the animator, frames-per-second, camera translation, etc.

Quicktime example: *previs_scene.mov*

### Preparing a scene for the stage

The animator has finished previz-ing the scene, and the director loves it. The character swoops heroically through the air, covering 10 city blocks in a matter of seconds. However, after the producer found out how much it costs to shut down 10 city blocks and fly an actor on wires from the world's largest crane, you've been told that the shot will in fact be done on a stage, with a green-screen backing.

### Zero-ing the character's motion

Because the camera is basically traveling with the character, the only information we need to preserve is the camera move *relative* to the character. Here's a quick way to do this:

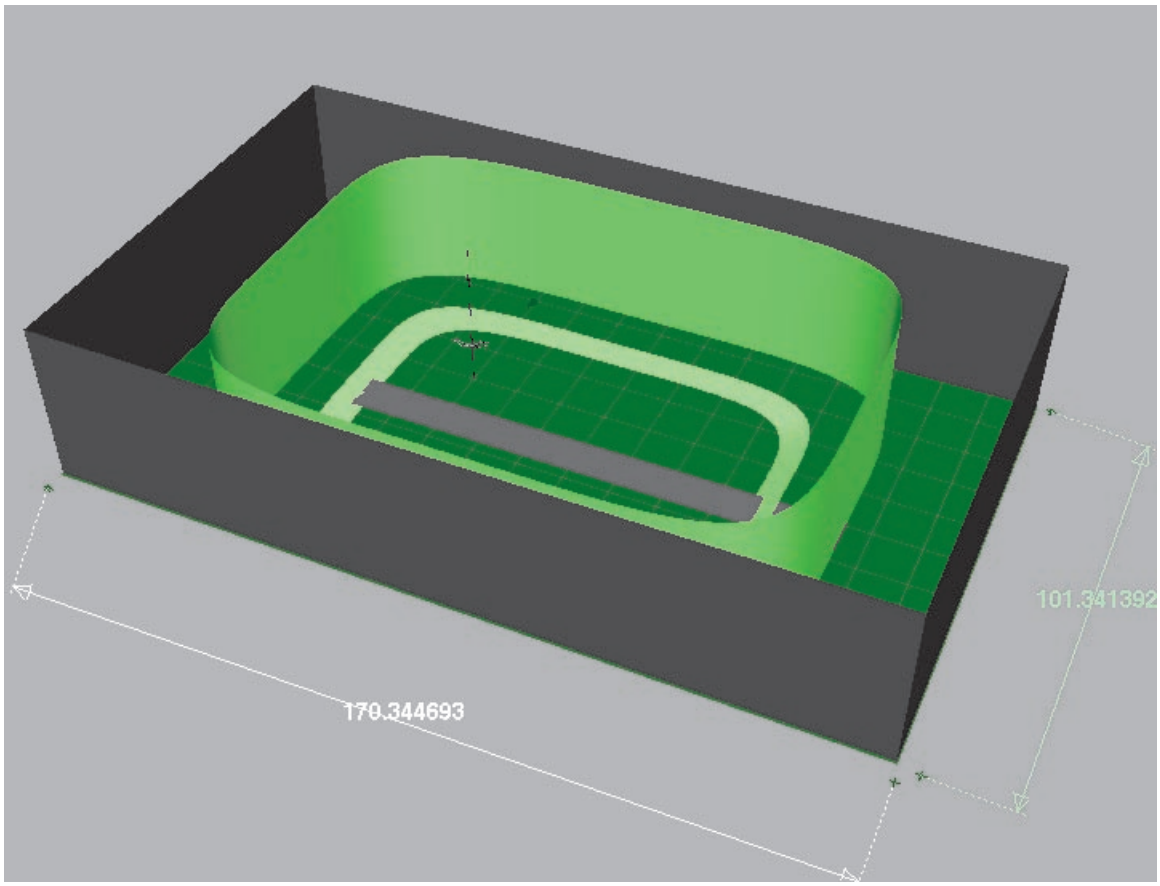
- Duplicate the shot camera.
- Make the new camera a child of the character.
- Point and Orient constrain the new camera to the original.
- Bake the translation and rotation channels of the new camera.
- Delete the translation channels of the character.

If you look through the new camera, you should see the same view of the character as before.

Quicktime example: *previs_stage.mov*

## Placing the animation in the stage

Hopefully the new camera move will fit within the dimensions of your stage. Having an accurate model of the stage is critical to previz, so ideally a survey of the stage should be done.




*Stage model with green-screen cove*

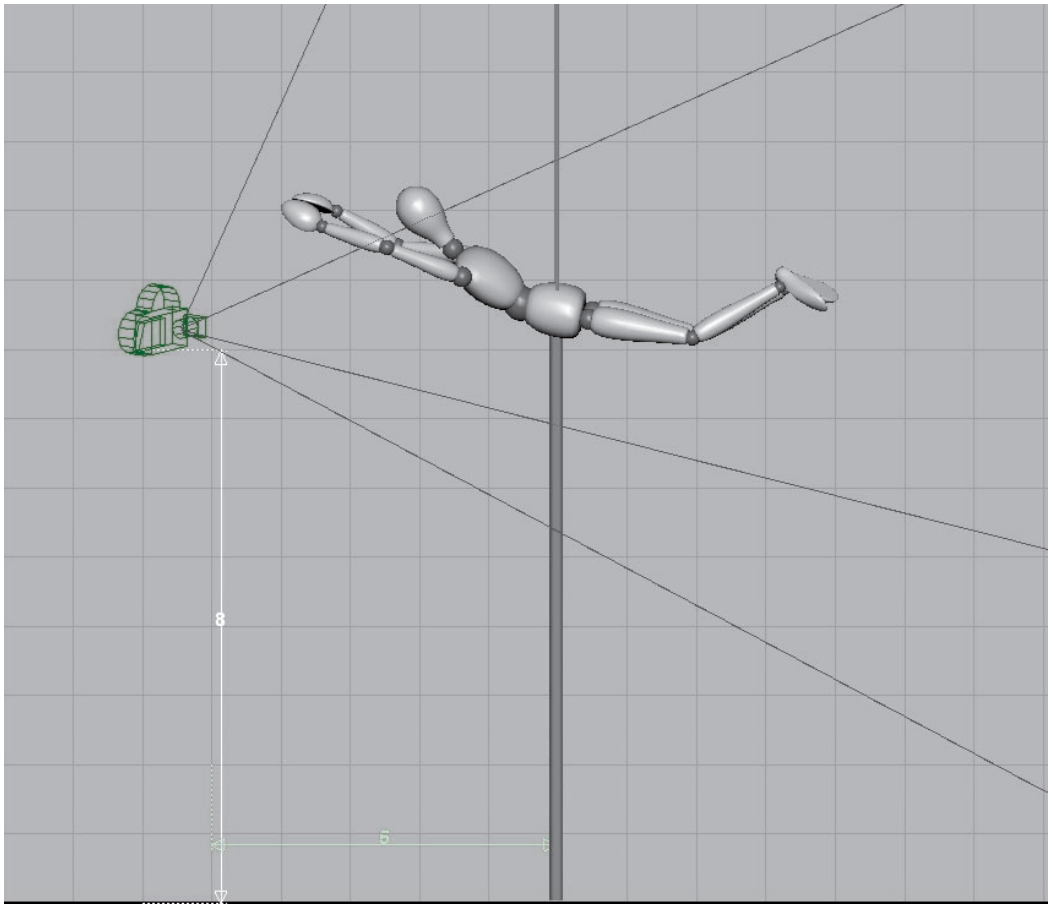
Since the character's overall translation has been erased, you can move it and the camera to any point in the stage.

## Rendering Scenes For Use On Set

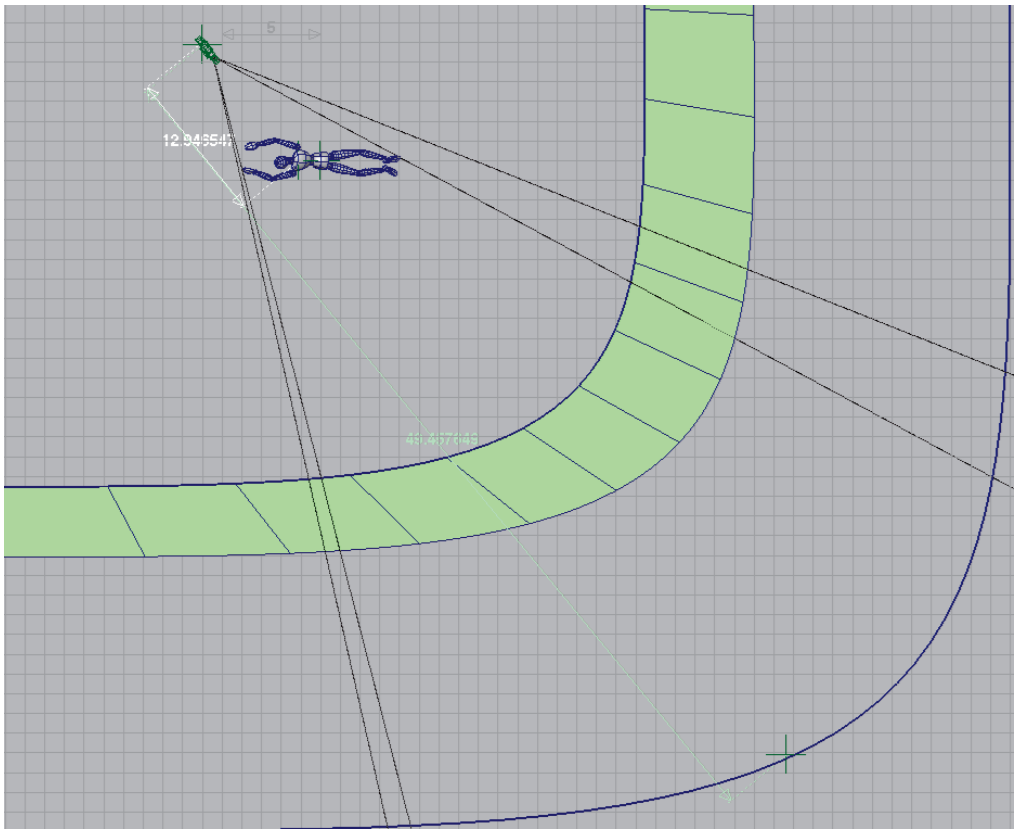
The more data you can provide to crew on set, the closer they will be able to match the previs. By playblasting the scene (with the heads-up display discussed earlier) and putting it on a tape or dvd, you can provide a quick reference.

If you're shooting against green-screen, it can be useful to playblast just the background without the foreground objects so they can be composited on-set. If you are shooting backgrounds which will have foreground objects added to them later, you can playblast previz foreground objects over green or blue for the on-set composite.

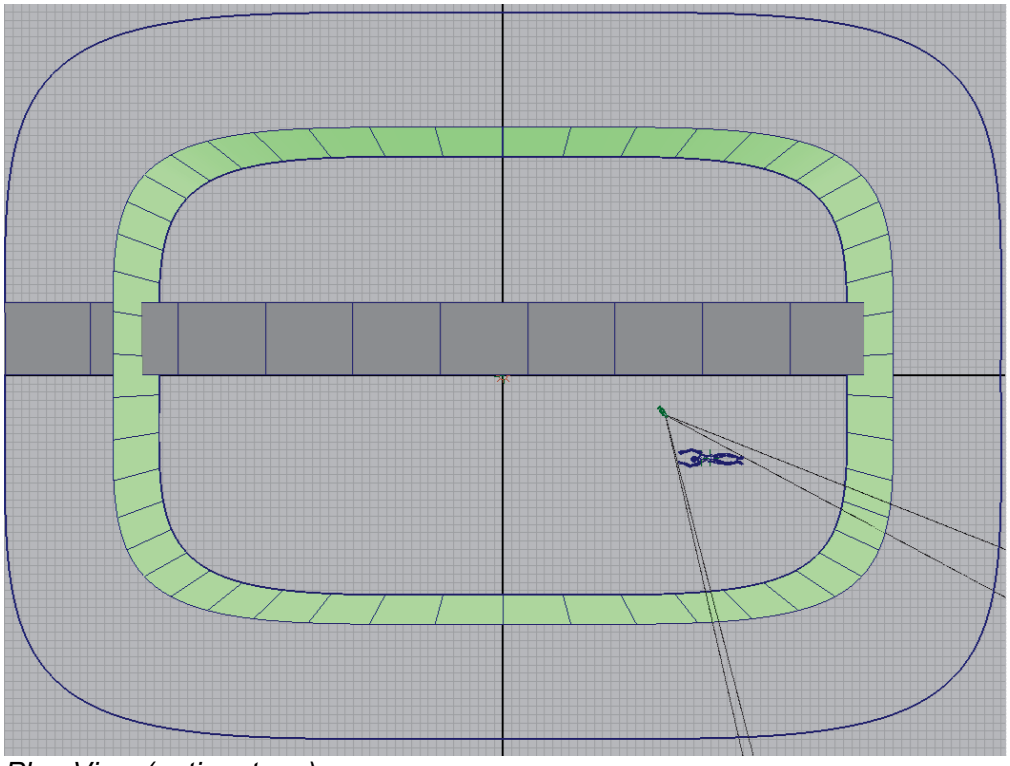
Hard copy prints of top and side views can be useful for quick reference on set. Be sure to include relevant measurements. You can use Maya's grid to show the scale of the scene. Turning on the camera's "show clipping plane" option gives an indication of the field of view. Here are some examples with a 1-foot grid:



*Side view print-out*



*Top view print-out*



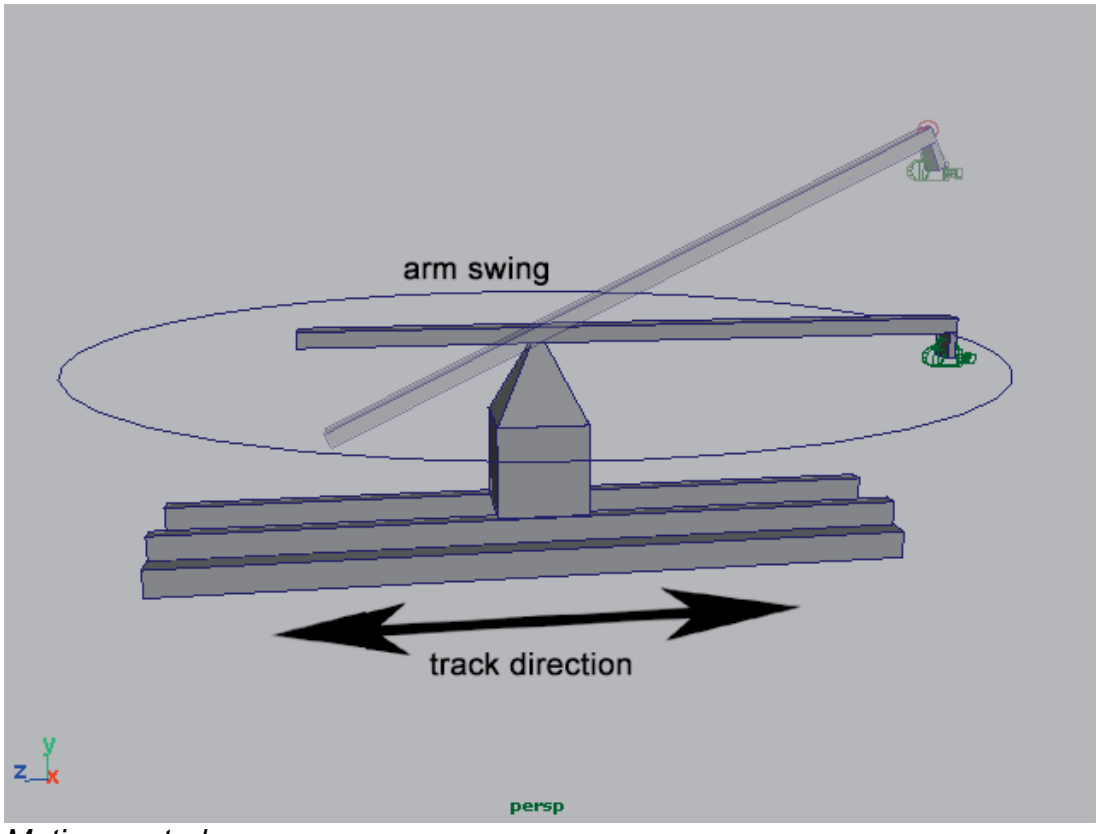
*Plan View (entire stage)*

## Motion Control

Many visual effects are shot with motion control cameras, allowing precise replication of the camera move. For instance, a background plate is shot, and then the exact camera move can be replicated later on a foreground element against a green-screen. There are many other uses for motion control, including “split screen” effects, crowd duplication, shooting miniature models, etc.

Another advantage of motion control is that the camera is controlled by a computer, so you can take data from Maya and send it directly to that computer. You can also import data from the moco computer into Maya, making it easy to track cg elements to the background. However, there are some limitations to motion control that need to be considered when previsualizing moco shots.

As with all elements of previz, it’s important to have an accurate model of the moco rig you’ll be using. Many moco companies already have models built for various 3d software packages, including Maya. In most cases they can also provide a plugin to solve the moco move based on an animated camera. Solving the move is the process of taking the cg camera and figuring out how the rig should move to match that camera as closely as possible. Unlike our cg camera, moco rigs have limited axes of movement. Here is an example of a moco crane




*Motion control crane*

- Crane base can move along the track at up to 6 feet per second

- Arm can swing 360 degrees around the base
- Arm can tilt +60 or – 30 degrees around the base
- Camera head can rotate nodally on all axes
- Different rigs have different limitations, make sure you get the specs from the moco company.

In addition to limitations of movement, limitations of speed must be considered. The crane will have a maximum speed that it can move along the track, as well as a maximum speed of rotation for the arm. Quick changes in direction can also be a problem. If you've previz'd a fast move, you may find it necessary to shoot the move slower, and reduce the frame rate of the camera to compensate.

**Transferring Data to and from Motion Control Cameras**

Ideally the motion control company can provide a plugin to take your scene directly from Maya to the moco rig. However, if no Maya plugin is available for that system, you will probably need to export kuper data. The kuper format is the most common language for motion control systems, and is relatively easy to export from Maya since it is a simple ascii format.

Example of Kuper file:

```
Axes = VTrack, VEW, VNS, VPan, VTilt, VRoll
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
2.0000 2.0000 2.0000 2.0000 2.0000 2.0000
3.0000 3.0000 3.0000 3.0000 3.0000 3.0000
```

1. Exporting Kuper from Maya:

   - Select the camera to export
   - Select **File > Export Selection > Option Box**
   - Set **File Type** to **move**
   - Select all the translate and rotate channels in the channel box
   - Click **From Channel Box**
   - Click **Export Selection** and save the file
   - Open the file in a text editor

The .mov file contains the channel information, but needs the axes label to be read as kuper. Add this line at the top of the file:

```
Axes = VEW, VNS, VTRACK, Vtilt, Vpan, Vroll
```

This is in the same order as Maya's default channel order.

| | | | | |
|---|---|---|---|---|
| VEW | = | Virtual East-West | = | translateX |
| VNS | = | Virtual North-South | = | translateY |
| VTRACK | = | Virtual Track | = | translateZ |
| Vtilt | = | Virtual tilt | = | rotateX |
| Vpan | = | Virtual Pan | = | rotateY |
| Vroll | = | Virtual Roll | = | rotateZ |

The kuper software calculates the rotation of the arm using an ik type solution, similar to Maya's ik solver.

## Miscellaneous Tools

Included on the cdrom are the following scripts we've developed for previs purposes:

- flipcam.mel flips orthographic view (top becomes bottom, left becomes right, etc.)
- mask.mel creates and assigns an image as an imageplane to a selected camera. Several images are provided for use as reticles and masks.
- modelClipPlanes.mel creates geometry that represents the clipping planes, so they are visible during a playblast. Can be useful to illustrate the field of view of the selected camera.
- previs_hud.mel creates several heads-up-display elements that show camera information for the scene.
- silhouette.mel toggles default lighting so the silhouette is easy to read.
- zoomCamera.mel duplicates the shot camera and uses the overscan to zoom into the imageplane, creating a close up view of the shot.
- cameraDummy.mel creates camera-shaped geometry and constrains it to a camera so it can be scaled independently and rendered.
- objectCam.mel creates a camera constrained to an object, useful for following a detail of a character while it's in motion.
- windowCamera.mel creates a floating window with a specified camera view.

### Installation

- Copy the contents of the scripts directory to your local scripts dir, e.g. "My Documents/maya/5.0/scripts".
- Copy the contents of the shelves directory ("shelf_previs.mel") to your local shelves directory (e.g. "My Documents/maya/5.0/prefs/shelves").
- Copy the contents of the icons directory to your local icons directory (e.g. "My Documents/maya/5.0/prefs/icons").
- Copy the contents of the images directory to the "sourceimages" directory of whichever project you'll be working in (e.g. "My Documents/maya/projects/default").

### Using the scripts

On the previs shelf there are 13 buttons:

1. creates a previz heads-up-display.
2. creates a 2.35 mask.
3. creates a 2.35 reticle.
4. creates a 1.85 mask.
5. creates a 1.85 reticle.
6. toggles default lighting off (silhouette)
7. divides the view into thirds
8. flips an orthographic view

9. creates a copy of the camera that can be zoomed via the overscan
10. creates geometry for the clipping plane
11. creates geometry for the camera that can be scaled
12. creates a camera constrained to an object
13. creates a floating window for a camera

All the mask and reticle images are designed to fit a 35mm full aperture film aspect ratio (4:3) cg camera, and will not be an accurate guide if used with a different filmback. previs_hud.mel, mask.mel, zoomCamera.mel, modelClipPlanes.mel, cameraDummy.mel, objectCamera.mel and windowCamera.mel all require a camera to be selected before running.