What Is a "Good" Calibration?

To answer this question, you need to consider the goals of matchmoving:

- Reproduce a 3D camera that matches the one used to film the scene
- Find the 3D locations of features in the scene that was filmed

Consider for a moment what is necessary to completely define a camera. We need to know its 3D position and rotation as well as its focal length, film back, principal point, and the amount of lens distortion. When you consider that the matchmoving program must do this for every frame in the sequence, it's no wonder it gets a little fussy when we try to solve a difficult shot!

Note: Throughout this book I use the terms *calibration* and *solution* somewhat interchangeably because they mean virtually the same thing. Technically speaking however, "achieving a calibration" refers only to cameras, whereas "achieving a solution" generally refers to the entire 3D scene.

In most matchmoving programs, the calibration is a one-button operation. You simply instruct the program that you want to solve the shot. The program analyzes the 2D tracking you have provided and (hopefully) generates a camera that matches the real-world camera. It also creates markers that represent the 3D locations of the features you tracked in 2D tracking. But how do you determine whether the camera calibration is good or bad?

Residuals

When a matchmove is solved correctly, the 3D camera matches the real-world camera exactly. The best way to tell whether or not the two cameras are identical is to look through them. The view afforded by the real-world camera is the image itself. The view from the new 3D camera is the 3D scene. If our calibration is correct, then the 3D markers should line up with the feature they represent in the image (Figure 4.1).

For example, let's say we placed a 2D track on the corner of a building. After calibration, that 2D track is converted to a 3D marker that represents the 3D position of the corner of that building. When we look at the 3D marker through the calibrated 3D camera, it should appear to line up with the original 2D track on the image. Therefore, a good calibration could be defined as the perfect (or near-perfect) alignment between the 2D track and the 3D marker when viewed through the 3D camera.

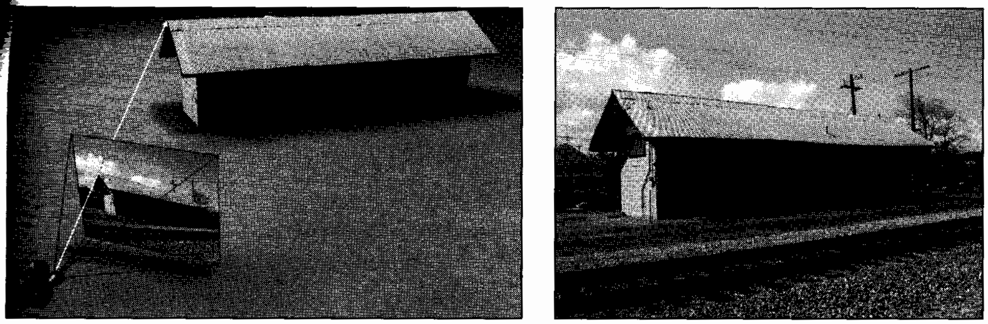

Figure 4.1 When the camera is correctly calibrated, you can draw a straight line between the lens center, the track on the 2D image, and the 3D marker in the 3D scene. The image on the right shows how the scene looks when viewed through the 3D camera.

The alignment of the 2D track and 3D marker is almost never perfect. The difference between them is called a residual (Figure 4.2). Residual values are usually expressed in pixels. For example, if the 3D track is reprojected onto the 2D image plane and ends up being 3 pixels away from the original 2D track, it is said to have a residual of 3. If the residual of the majority of the tracks is less than 1 pixel for the duration of the shot, it is considered a good calibration. Why 1 pixel? If the shot is tracked at full resolution, any track that slips by less than a pixel shouldn't be noticeable.

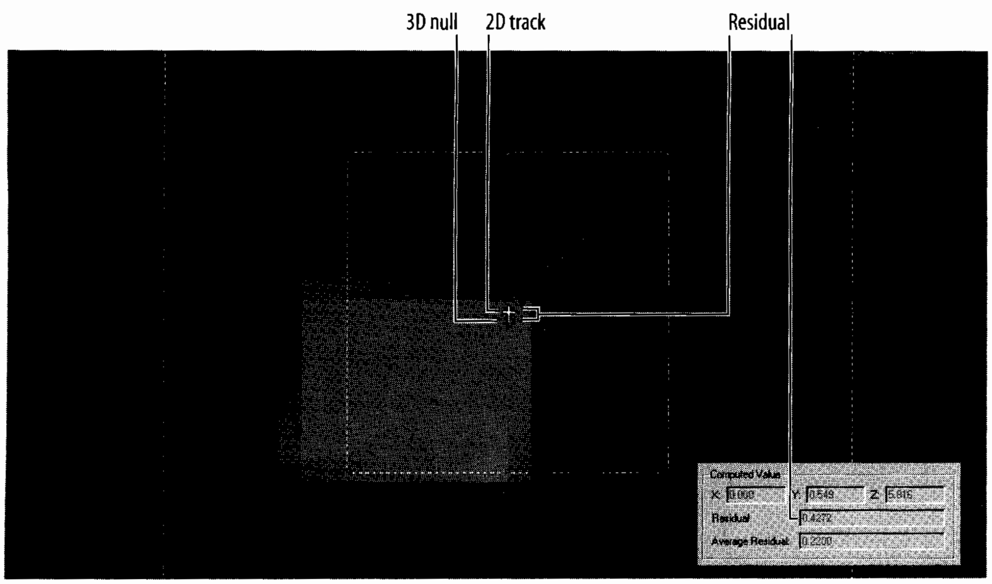

Figure 4.2 A residual is the difference between the 2D track and the 3D marker as seen through the camera. The top crosshair represents the 2D track and the bottom one the 3D marker. The difference between them in pixels is the residual—in this case, about a 0.4 pixel difference.

Calibrating Your Cameras

I've heard that there's not much room for aesthetics in matchmoving because the solution either works or it doesn't. Perhaps it's technical nature of this discipline or a lack of understanding of the process of matchmoving that gives people this impression. But there is an art to getting the right solution, especially the difficult ones. Whatever the reason, knowing how a calibration happens can be a great asset to a matchmover.

A matchmove doesn't simply pop into existence. Like many other aspects of 3D effects, such as animation or lighting, matchmoving can be an iterative process where a solution must be zeroed in on rather than solved the first try. Many times, you will have a partial solution or a "weak" (inaccurate) solution that must be improved. Sometimes you'll fight the nature of the footage (pans, zooms, etc.), and other times you'll deal with your own bad 2D tracking. But whatever the cause, you can take heart in knowing that most matchmoves need a little extra "love" before they're ready to deliver.

In order to understand why matchmoves can fail, we need to look at how matchmoves are solved by the matchmoving program. No, I won't bombard you with matrix notation and trigonometry. I simply mean we're going to look at how the software goes about doing its job. A lot of what happens during calibration is invisible to the user. In this section, we'll look at what happens "under the hood" while the matchmove program is solving the scene.

Finding the Right Fit

When beginning a calibration, the task at hand is rather enormous. There are an infinite number of possible camera/track/null configurations but only a few that will produce the desired results—a perfect alignment of all three elements during the entire shot. Matchmoving software tackles this problem by breaking the task up into smaller, more manageable subtasks.

The program usually begins its calibration by scanning the 2D tracks available and making a very cursory examination to decide on the frames that will be used as a starting point. These frames, which I'll call the "initial frames," are calibrated by themselves and used to generate a camera and nulls (Figure 4.3). This minicalibration becomes the baseline for the full calibration. The software starts adding more and more frames throughout the shot to its initial calibration and makes slight adjustments to the overall solution as it does so. As a final step, the software adds in all of the in-between frames to the solution. In this manner, the software can start with a rough calibration and refine it until it achieves a single solution that works for all frames.

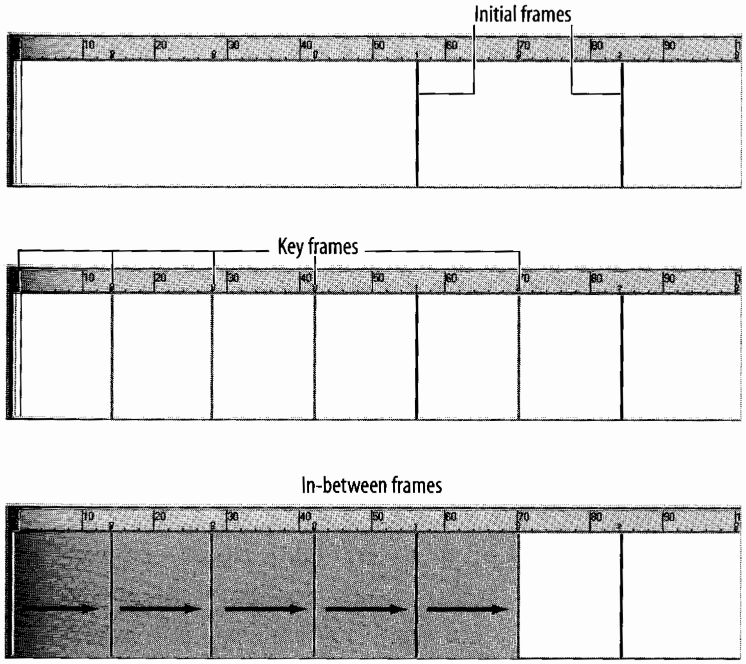

Figure 4.3 The matchmoving program begins by calibrating just a small number of frames, then it adds more frames to the solution and adjusts as necessary. Finally, the program adds in all of the in-between frames to the solution and creates a single solution that works for all frames.

The matchmove program uses an iterative process for determining the camera location, and it's a good reason to approach your calibration the same way. For example, if you find problems in a certain area of the shot, it would make sense to start troubleshooting in that area. We'll discuss troubleshooting later in this chapter and again in Chapter 10, "Troubleshooting and Advanced Techniques," but suffice it to say that you should expect to work *toward* a solution rather than instantly arriving at one. Consequently, the first skill you must master is how to evaluate the matchmove as you work.

Calibration Tutorial

To illustrate how a basic calibration works, we'll use the tutorial from the previous chapter. We'll continue using MatchMover Pro 3.0 for this calibration, but the techniques are similar in other matchmove programs. If you haven't saved your scene, you can use the project named `bluescreen_03_2DTrackDone.mmf` that is included in the Chapter 3 folder on the CD.

Normally, before we run the calibration, we would give MatchMover Pro any information that we have. For now, let's assume we don't have any information on the scene. We don't, for example, know anything about the camera or what focal length was used. There are, however, a few assumptions we can make about these values.

It seems fairly obvious that the camera is not doing any zooming during the shot, and this is one important piece of information we can use during calibration. MatchMover Pro doesn't necessarily need to know the exact focal length in order to achieve a solution; it merely needs to know what type of lens was used.

1. It's always a good idea to check your camera settings before you begin calibration. To do this, click the plus sign on the Cameras folder in the Project View on the left side of the interface, then click again on the Camera. This opens up the Properties Window for the camera on the right (Figure 4.4).

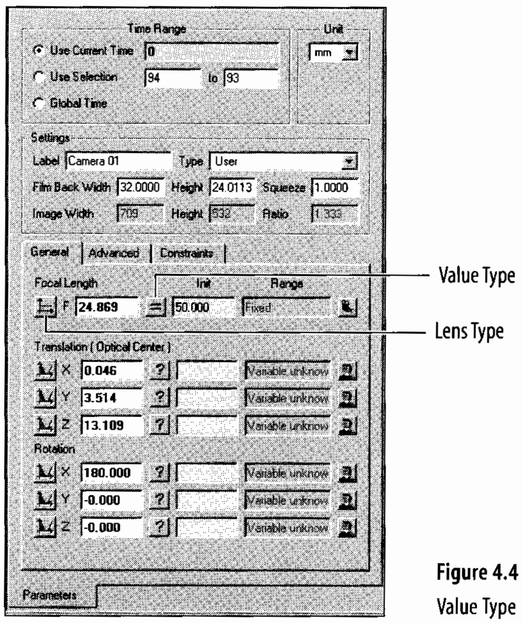

Figure 4.4 The camera parameters. The Lens Type and Value Type buttons allow you to set the lens parameters.

The two fields you will be most concerned with are the focal length field and the film back. We don't know anything about this camera, and therefore we don't know the film back. So for now, we'll let the program use the default film back size, which is approximately 32.00mm × 24.00mm.

2. As for the focal length, there are two considerations: the lens type and the value type. The Lens Type button sets what type of lens was used: choose Constant.

Clicking the first button in the Focal Length row toggles the setting between Constant or Variable:

Constant The lens doesn't zoom or change focal length during the shot.

Variable The lens zooms during the shot or is a variable lens.

Note: The film back and the focal length work together to show the proper view of the scene. When the precise value of the focal length is required, you must also know the precise film back size. We can ignore the film back measurements in this case because we are not trying to solve for a specific focal length. The program will automatically compensate for any film back differences by changing the value of the focal length to a value that will work.

3. The next field is where you would set the focal length if you knew it, as well as where any solved focal lengths would appear. After that is another small button that sets the Value Type. It toggles between two types, Fixed or Initialized; choose Initialized.

Fixed You know the exact value and will set that value in the Init field (to the right of the button). You generally want to use this type if you know the exact value of the focal length.

Initialized You do not know the exact value. When you are using this value type, you can enter the approximate value in the Init field if you have any idea what it might be. The program then uses that value as its starting point for the calibration but may change the value if it needs to.

In our tutorial, we'll need to use the Constant/Initialized setting. This is because we don't know what the focal length is, but we do know that it's constant.

Once you've set the camera's parameters, you're ready to run the calibration—the moment of truth!

4. Under 3D Tracking, choose Solve for Camera or hit the F9 key. MatchMover Pro begins working through the calibration process. How fast this goes depends on your processor speed, but it shouldn't take more than a minute or so. As the program works, I like to keep my eye on the status bar at the lower-right side of the frame. This lists the residual values of the tracks as they are solved. When a calibration is going well, these values will generally stay under 1.0, and when a solution is more difficult, these values may creep up over 1.0. If the values are under 10 or so, you may have problems but probably correctable ones. If the values get up into the hundreds or even thousands, you may want to hit Escape to stop the tracking process, because something is obviously wrong.

After the calibration is complete, MatchMover Pro will give you some initial clues as to how well the cameras are calibrated. The most important indicator is visible at the very top of the Timeline (Figure 4.5). You'll see a colored bar beneath the frame numbers, with key icons along the top of it. This shows you how well the camera has solved. If it has solved correctly, it will be all green. Any yellow or red here is an indication that the camera wasn't solved well.

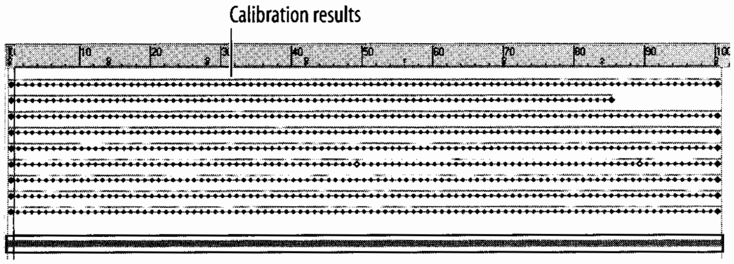

Figure 4.5 The very top of the Timeline serves as an indicator of the quality of the solution.

> **Note:** In the preceding chapter, I said it was usually not a problem if the track's individual colors contained some red or yellow. This is not the same for the colored bar at the top of the Timeline, which indicates the status of the matchmove. The individual track's color has to do with the tracker's confidence in how well the pattern compares from one frame to the next. The bar at the top of the Timeline is the average of all the track's residuals. If the bar is red, that means that enough tracks are misaligned and the matchmove is bad during those frames.

If everything in your Timeline and Project View looks green, then you probably have a good solution. I say *probably* because sometimes MatchMover Pro indicates it has a good solution, but in fact it may not. The program may have found a solution that works mathematically, but not in a sense that is useful to us. In the next section, we'll evaluate the solution a little more thoroughly to make sure we have a good solution.

Evaluating the Solution

As you work on a matchmove and generate solutions, you will need to verify that they are, in fact, valid solutions. One common beginner's mistake is to create a solution and immediately export the solution to the 3D scene. The problem is, if the matchmove isn't right, you'll have to go back to the matchmoving program and fix the solution.

A more efficient way to work is to use a step-by-step analysis during each stage of the process. While performing your final checks in the matchmoving program, you should evaluate:

- 3D nulls
- 3D space
- The rendered movie

Evaluating the 3D Nulls

The first stage of evaluation is to examine the results in the matchmoving software. The first test I like to perform is to compare the 2D and 3D markers. To do this, you need to look at the nulls through the camera by clicking the small 3D button in the upper-left corner of the Video Window, then right-clicking and choosing Lock to Camera. This allows you to see the nulls directly over the features they represent (Figure 4.6). The nulls should appear to match up with the features that were originally tracked.

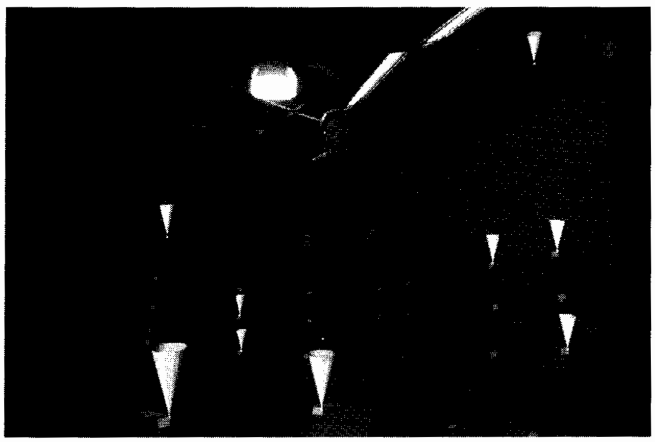

Figure 4.6 The first item to check is whether or not the 3D markers seem to line up with the correct features on the image.

The 3D markers can also help you determine the quality of the solution in another way. If your solution is accurate, then the size of the nulls should reflect this. For example, nulls that are closer to the camera should appear larger, while nulls that are farther away will be smaller. Notice in Figure 4.6 how the nulls on the back wall seem much smaller than the ones on the foreground boxes. This is what we would expect to see for any objects placed in the scene.

If this seems to be the opposite of what you're seeing through your camera, you may have a problem with the solution. It is fairly common to have one or two points that do not seem to be quite the right size, but if the discrepancy is happening globally, investigate further.

Evaluating the 3D Space

If the 2D/3D track comparison looks good, then proceed to the 3D perspective view and look at the camera and nulls. Does the camera move as you would expect it to? Are there any obvious spikes in the movement? Is there an excessive amount of jitter or noise in the camera path? Are the 3D nulls where you would expect them to be? Are any markers missing? Common signs of a failed solution (Figure 4.7) would include:

- Camera path appears random or erratic.
- Camera path works for one portion of the sequence then jumps to another location before finishing its move.
- Camera is not visible at all.
- Nulls are in the wrong position (i.e., nulls that should be far away from the camera are close to it and vice versa).
- Nulls are all in a line, on a single plane, all crammed into a single point, or even behind the camera.
- Nulls are not visible at all.

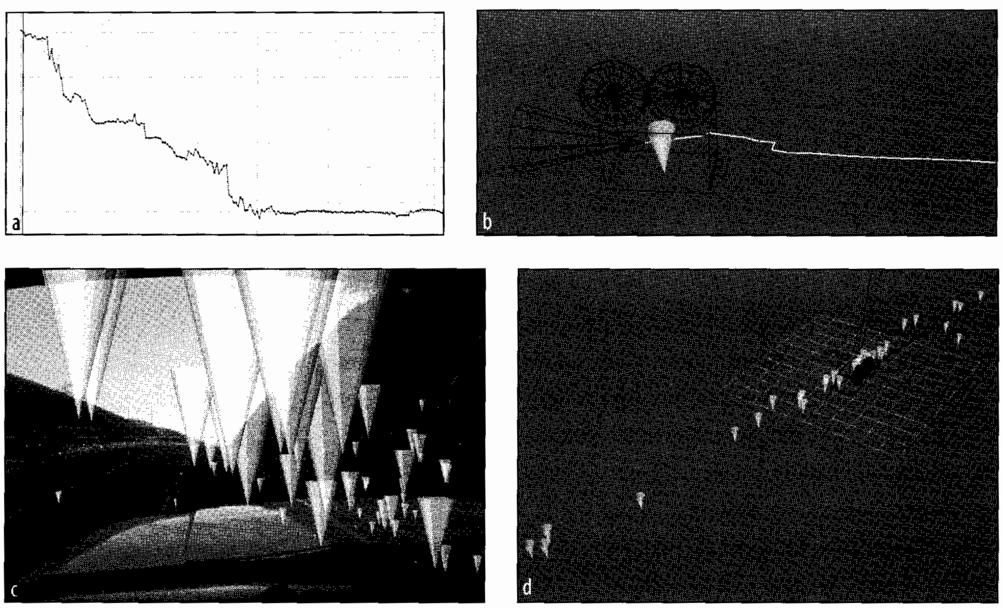

Figure 4.7 Some common signs of a failed solution: (a) erratic motion or motion curves, (b) broken motion path, (c) 3D markers are not appropriate scale, and (d) 3D markers are in a line or behind the camera.

When you are looking at your 3D scene and any of these situations is present, it's possible that your solution is not perfect. That's not to say it's hopeless. In the case of certain difficult shots, an imperfect solution may be better than no solution at all. You might get one-half of the shot to solve correctly, or maybe the nulls aren't spaced properly, but perhaps the camera moves more or less the right way.

Note: In Chapter 7, "Integrating Matchmoves with the Scene," I discuss ways to manipulate imperfect solutions in the 3D application, but there are plenty of things you can do in the matchmove software to help fix or improve bad solutions. I discuss ways of doing this more in an upcoming section on troubleshooting, "When Good Solutions Go Bad."

Rendering the Matchmove

Before you export your final solution, it's a good idea to do one final check in the matchmoving program. So far we've examined the 3D space in the perspective view and looked at the markers over the image. But there are some problems you might only notice when the image sequence is playing, such as:

- High-frequency noise
- Subtle drift
- Sudden pops in the camera's rotation

Most matchmoving programs include a way to render a test movie to see whether the markers are in fact sticking throughout the sequence. It basically amounts to rendering the matchmove tracks as seen in the 3D view.

To illustrate how this works, let's render out a test of our calibration for this shot:

1. Make sure you are in the 3D view in MatchMover Pro. If you aren't, click the small 3D button in the upper-left corner of the Video Window. You also need to be "locked" to the camera, that is, the view of the markers is the same as if you were looking through the calibrated cameras. To do this, right-click in the Video Window and choose Lock on Camera. This should give you a view of the plate with the 3D markers on top of it.
2. Choose 3D Scene > Render Setup (Figure 4.8). This is where you set up all the parameters for your render from MatchMover Pro. You will probably want to render the full length of the clip at 100% size. You can choose a file type and use a compressor if you'd like, but don't compress it so much as to make it visually difficult to see the markers. You should also render out the markers as Anti-Aliased, which will make them smoother.

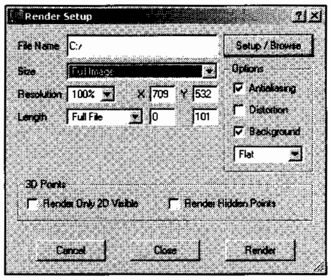

Figure 4.8 The Render Setup panel

3. Click the Render button. You'll see MatchMover Pro start working its way through the render. When it's finished, it will play the video back and provide a way for you to scrub the footage to check for problems.

When Good Solutions Go Bad

It's entirely possible that your calibration may not have turned out well. Fortunately, with this type of shot, there are only a few things that could go wrong. I cover troubleshooting in detail in Chapter 10, but here are a few things that could conceivably derail your calibration in this shot:

Slipping tracks I've found that most problems with a calibration lead back to the 2D tracking. In this particular shot, the features should be easy to track, and there are plenty of markers on the wall in various areas throughout the shot. Make sure that none of your tracks are slipping, especially as they approach the edge of the frame. An easy way to do this is to select a track and then right-click in the Video Window and choose Lock on Track. This keeps the track always centered in the window. You can then scrub the video by holding down the Ctrl key and dragging back and forth in either the Video Window or the Timeline.

Tracks not properly spaced Make sure that your tracks are well spaced throughout the environment and not all bunched up in one area or only on the walls. Remember, you're trying to sample the 3D space. You need to help MatchMover Pro understand where the walls, cubes, and ceiling are through 2D tracking.

Not enough tracks MatchMover Pro requires that you have at least eight tracks on the screen at any given time. If one of your tracks leaves the screen, make sure another replaces it before the departing one exits the frame.

Tracking a moving object All the features you track should be static. In this shot, that's pretty easy. The only things you should *not* track are the man, that little moving marker on the foreground cube, and the hood on the camera.